Differential Privacy for Relational Algebra: improving the sensitivity bounds via constraint systems

Marco Stronati
Università di Pisa, Italy
marco@stronati.org

Catuscia Palamidessi
INRIA and LIX, Ecole Polytechnique, France
catuscia@lix.polytechnique.fr

Statistical Disclosure Control

Revealing accurate statistics

vs

Preserving the *privacy* of individuals.

Statistical Disclosure Control

Revealing accurate statistics

vs

Preserving the *privacy* of individuals.

How many people have cancer?

Statistical Disclosure Control

Revealing accurate statistics

vs

Preserving the *privacy* of individuals.

How many people have cancer?

Does John Doe have cancer?

Information Hiding

Dalenius' *ad omnia* privacy desideratum ('77):

> *nothing about an individual should be learnable from the database that could not be learned without access to the database.*

Information Hiding

Dalenius' *ad omnia* privacy desideratum ('77):

nothing about an individual should be learnable from the database that could not be learned without access to the database.

Trade off between *privacy* and *utility*

Information Hiding

Dalenius' *ad omnia* privacy desideratum ('77):

nothing about an individual should be learnable from the database that could not be learned without access to the database.

Trade off between *privacy* and *utility*

Quantitative Approach

Differential Privacy - Dwork, McSherry, Smith, Nissim

A randomized function $\mathcal{H} : \mathcal{R} \to \mathbb{R}$ satisfies ϵ-differential privacy if for all pairs $R, R' \in \mathcal{R}$, with $R \sim R'$, and all $X \subseteq \mathbb{R}$:

$$Pr[\mathcal{H}(R) \in X] \leq Pr[\mathcal{H}(R') \in X] \cdot e^{\epsilon}$$

Differential Privacy - Dwork, McSherry, Smith, Nissim

A randomized function $\mathcal{H} : \mathcal{R} \to \mathbb{R}$ satisfies ϵ-differential privacy if for all pairs $R, R' \in \mathcal{R}$, with $R \sim R'$, and all $X \subseteq \mathbb{R}$:

$$e^{-\epsilon} \leq \frac{Pr[\mathcal{H}(R) \in X]}{Pr[\mathcal{H}(R') \in X]} \leq e^{\epsilon}$$

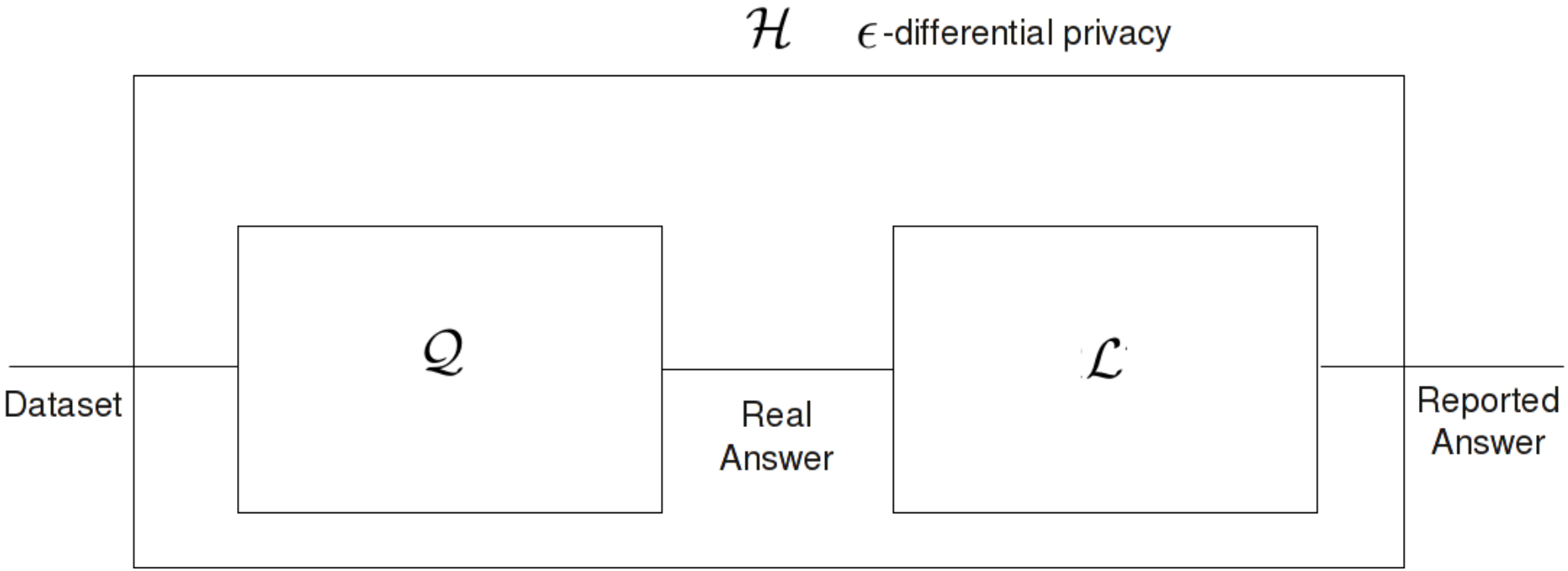
Differential Privacy - Dwork, McSherry, Smith, Nissim

A randomized function $\mathcal{H} : \mathcal{R} \to \mathbb{R}$ satisfies ϵ-differential privacy if for all pairs $R, R' \in \mathcal{R}$, with $R \sim R'$, and all $X \subseteq \mathbb{R}$:

$$e^{-\epsilon} \leq \frac{Pr[\mathcal{H}(R) \in X]}{Pr[\mathcal{H}(R') \in X]} \leq e^{\epsilon}$$

ϵ-indistinguishability

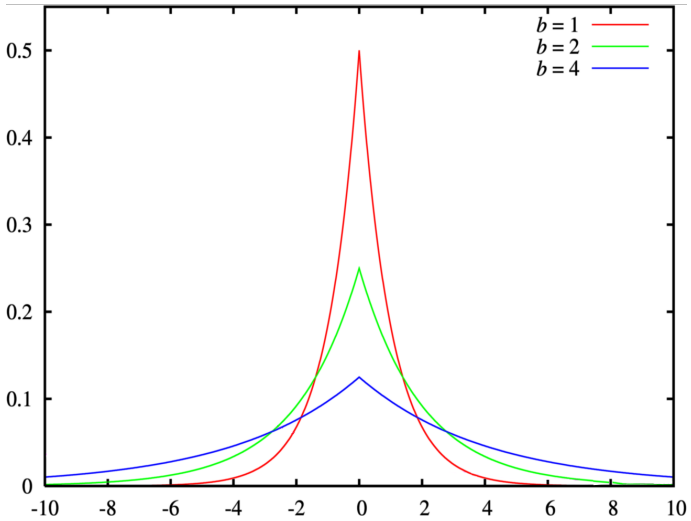
Overview (oblivious case)



Noise addition

Laplacian distribution

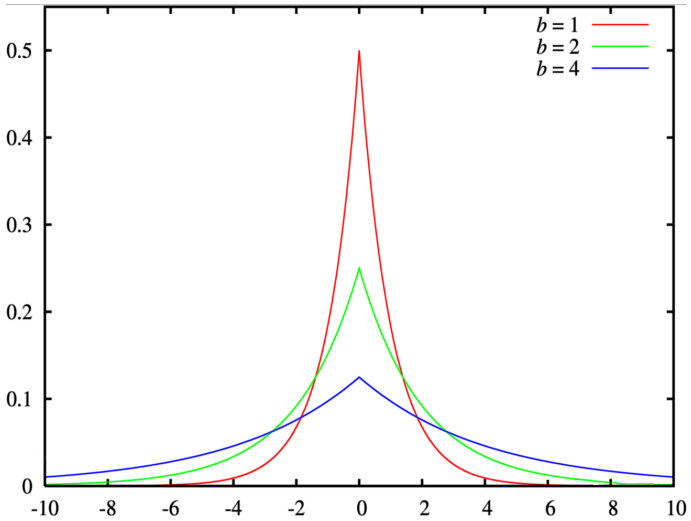
$$Lap(x \mid b) = \frac{1}{2b} exp\left(-\frac{|x|}{b}\right)$$



Noise addition

Laplacian distribution

$$Lap(x \mid b) = \frac{1}{2b} exp\left(-\frac{|x|}{b}\right)$$



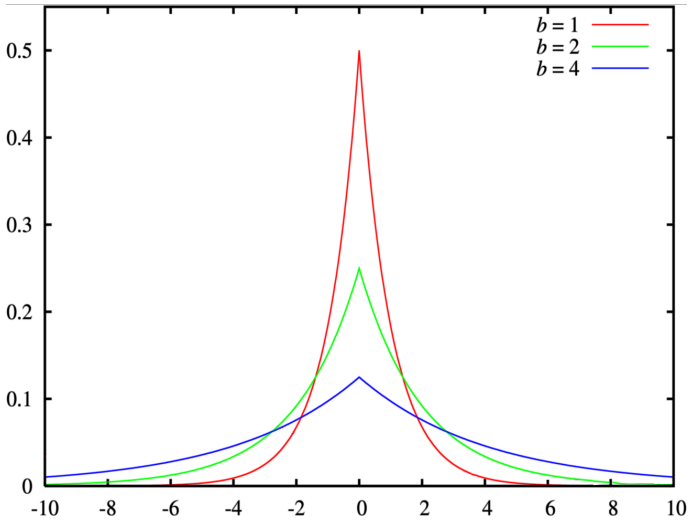
Theorem (Dwork06)

For $Q : \mathcal{R} \to \mathbb{R}$, the randomized mechanism $\mathcal{H}$ that adds noise with distribution $Lap(\Delta_Q/\epsilon)$ enjoys ϵ-differential privacy.

Noise addition

Laplacian distribution

$$Lap(x \mid b) = \frac{1}{2b} exp\left(-\frac{|x|}{b}\right)$$



Theorem (Dwork06)

For $Q : \mathcal{R} \to \mathbb{R}$, the randomized mechanism $\mathcal{H}$ that adds noise with distribution $Lap(\Delta_Q/\epsilon)$ enjoys ϵ-differential privacy.

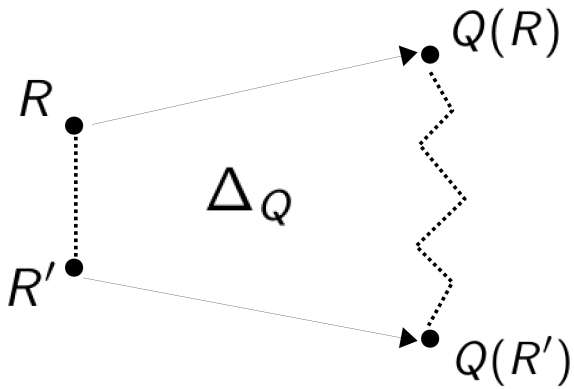
$$\uparrow \Delta_Q \qquad \downarrow \epsilon$$

Sensitivity

Definition (Sensitivity Dwork06)

Given a query $Q : \mathcal{R} \rightarrow \mathbb{R}$, the sensitivity of Q, denoted by Δ_Q, is defined as:

$$\Delta_Q = \sup_{R \sim R'} | Q(R) - Q(R') |$$



Contribution

- a compositional method to compute a bound on the sensitivity of a query expressed in relational algebra

Contribution

- a compositional method to compute a bound on the sensitivity of a query expressed in relational algebra
- constraints used to obtain the *exact* sensitivity

Differential Privacy for Relational Algebra

Relational Algebra - A Formal SQL

- $\mathcal{T}$: universe of tuples
- **Relation** R: a *set* of tuples
- $\mathcal{R}$: universe of relations

Definition (Relation Schema)

$$name(a_1 : D_1, a_2 : D_2, \ldots, a_n : D_n)$$

Relational Algebra - A Formal SQL

- $\mathcal{T}$: universe of tuples
- **Relation** R: a *set* of tuples
- $\mathcal{R}$: universe of relations

Definition (Relation Schema)

$$name(a_1 : D_1, a_2 : D_2, \ldots, a_n : D_n)$$

Example

```
Items  {  Item   :  String,
          Price  :  Int,
          Cost   :  Int      }
```

Item	Price	Cost
Oil	100	10
Salt	50	11

Constraints

```
CREATE   TABLE products (
         product_no integer,
         name text,
         price numeric CHECK (price > 0)
);
```

Constrained Schema

$$\mathcal{T}(C) \qquad \mathcal{R}(C) = 2^{\mathcal{T}(C)}$$

$$\texttt{Items} \quad \{ \begin{array}{lcl} \texttt{Item} & : & \texttt{String}, \\ \texttt{Price} & : & \texttt{Int}, \\ \texttt{Cost} & : & \texttt{Int} \end{array} \} \qquad \{ \begin{array}{rcl} 0 & < & \texttt{Cost} \\ \texttt{Cost} & \leq & \texttt{Price} \end{array} \begin{array}{l} \leq 1000 \\ \leq 1000 \end{array} \}$$

c-schema: schema + set of constraints C

Constrained Schema

$$\mathcal{T}(C) \qquad \mathcal{R}(C) = 2^{\mathcal{T}(C)}$$

$$\texttt{Items} \quad \{\ \texttt{Item} : \texttt{String},\ \texttt{Price} : \texttt{Int},\ \texttt{Cost} : \texttt{Int}\ \} \qquad \{\ 0 \leq \texttt{Cost} \leq 1000,\ \texttt{Cost} \leq \texttt{Price} \leq 1000\ \}$$

c-schema: schema + set of constraints C

Transformation from c-schema to c-schema

Sensitivity Constrained

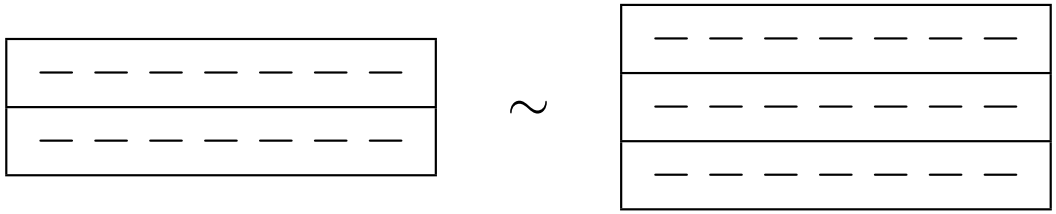
Definition (Sensitivity constrained)

Given $f : (X, d_X) \to (Y, d_Y)$, set of constraints C on X

$$\Delta_f(C) = \sup_{\substack{x,x' \in sol(C) \\ x \neq x'}} \frac{d_Y(f(x), f(x'))}{d_X(x, x')}$$

Metric Spaces

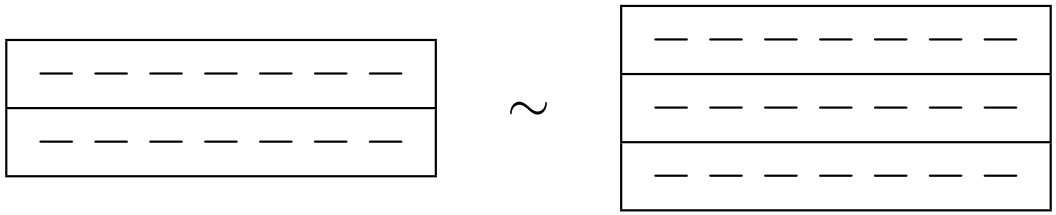
Adjacency relation $(\mathcal{R}, \sim)$



Hamming Graph

Metric Spaces

Adjacency relation $(\mathcal{R}, \sim)$

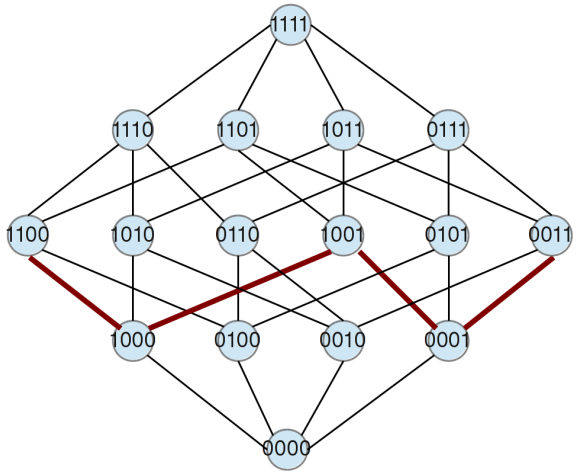


Hamming Graph

Definition (Hamming distance d_H)

Given $R, R' \in \mathcal{R}$

$$d_H(R, R') = |R \ominus R'| = |(R \setminus R') \cup (R' \setminus R)|$$



Metric Spaces

Definition (n-Hamming Distance d_{nH})

Given $\overline{R}, \overline{R}' \in \mathcal{R}^n$:

$$d_{nH}(\overline{R}, \overline{R}') = \max\left(d_H(R_1, R_1'), \ldots, d_H(R_n, R_n')\right)$$

Metric Spaces

Definition (n-Hamming Distance d_{nH})

Given $\overline{R}, \overline{R}' \in \mathcal{R}^n$:

$$d_{nH}(\overline{R}, \overline{R}') = \max\left(d_H(R_1, R_1'), \ldots, d_H(R_n, R_n')\right)$$

Definition (Euclidean Distance d_E)

Given $x, x' \in \mathbb{R}$

$$d_E(x, x') = |x - x'|$$

Structure of a Query

$$(\mathcal{R}^n, d_{nH}) \xrightarrow{\texttt{Op}} \ldots \xrightarrow{\texttt{Op}} (\mathcal{R}^n, d_{nH})$$

Structure of a Query

$$(\mathcal{R}^n, d_{nH}) \xrightarrow{\texttt{Op}} \ldots \xrightarrow{\texttt{Op}} (\mathcal{R}^n, d_{nH}) \xrightarrow{A\gamma_F} (\mathbb{R}, d_E)$$

Structure of a Query

$$(\mathcal{R}^n, d_{nH}) \xrightarrow{\mathtt{Op}} \ldots \xrightarrow{\mathtt{Op}} (\mathcal{R}^n, d_{nH}) \xrightarrow{A\gamma_F} (\mathbb{R}, d_E)$$

$$\mathtt{op} \quad \in \quad \begin{cases} \cup, \cap, \setminus, \sigma_\varphi \\ \\ \pi, \times, \bowtie \end{cases}$$

Operators Sensitivity

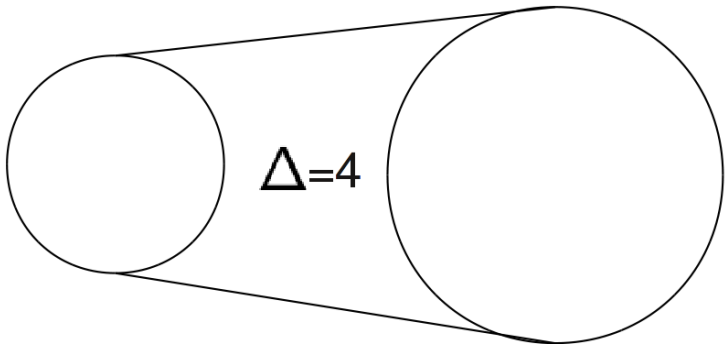
$$\Delta_{\mathrm{op}}(C) = \sup_{\substack{R,R' \in \mathcal{R}(C) \\ R \neq R'}} \frac{d_H(\mathrm{op}(R), \mathrm{op}(R'))}{d_H(R, R')}$$

Operators Sensitivity

$$\Delta_{\text{op}}(C) = \sup_{\substack{R,R' \in \mathcal{R}(C) \\ R \neq R'}} \frac{d_H(\text{op}(R),\text{op}(R'))}{d_H(R,R')} = \min\left(\Delta_{\text{op}}(\emptyset), \mathit{diam}(C \otimes C_{\text{op}})\right)$$

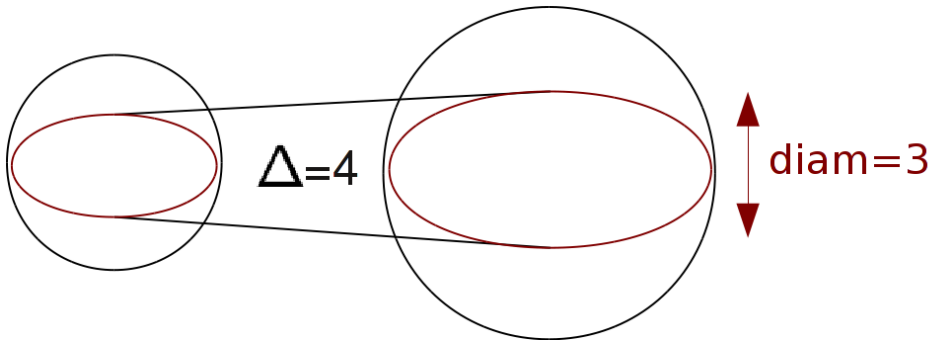
Operators Sensitivity

$$\Delta_{\text{op}}(C) = \sup_{\substack{R,R' \in \mathcal{R}(C) \\ R \neq R'}} \frac{d_H(\text{op}(R), \text{op}(R'))}{d_H(R,R')} = \min\left(\Delta_{\text{op}}(\emptyset), \mathit{diam}(C \otimes C_{\text{op}})\right)$$



Operators Sensitivity

$$\Delta_{\text{op}}(C) = \sup_{\substack{R,R' \in \mathcal{R}(C) \\ R \neq R'}} \frac{d_H(\text{op}(R), \text{op}(R'))}{d_H(R,R')} = \min\left(\Delta_{\text{op}}(\emptyset), \mathit{diam}(C \otimes C_{\text{op}})\right)$$



Other operators

op	Δ	*Schema*
$\cup$	2	$(A, C_1 \vee C_2)$
$\cap$	2	$(A, C_1 \wedge C_2)$
$\setminus$	2	$(A, C_1 \wedge (\neg C_2))$
σ_φ	1	$(A, C \wedge \varphi)$
$\pi_{A'}$	1	(A', C)

Cartesian product

$\times : (\mathcal{R}^2, d_{2H}) \to (\mathcal{R}, d_H)$

Name	Age	Height
John	30	180
Alice	45	160

$\times$

Car	Owner
Fiat	Alice
Ford	Alice

$=$

Name	Age	Height	Car	Owner
John	30	180	Fiat	Alice
John	30	180	Ford	Alice
Alice	45	160	Fiat	Alice
Alice	45	160	Ford	Alice

Cartesian product

$\times : (\mathcal{R}^2, d_{2H}) \to (\mathcal{R}, d_H)$

Name	Age	Height
John	30	180
Alice	45	160

$\times$

Car	Owner
Fiat	Alice
Ford	Alice

$=$

Name	Age	Height	Car	Owner
John	30	180	Fiat	Alice
John	30	180	Ford	Alice
Alice	45	160	Fiat	Alice
Alice	45	160	Ford	Alice

The (unrestricted) cartesian product has *unbounded* sensitivity.

Cartesian product

$\times : (\mathcal{R}^2, d_{2H}) \to (\mathcal{R}, d_H)$

Name	Age	Height
John	30	180
Alice	45	160

$\times$

Car	Owner
Fiat	Alice
Ford	Alice

$=$

Name	Age	Height	Car	Owner
John	30	180	Fiat	Alice
John	30	180	Ford	Alice
Alice	45	160	Fiat	Alice
Alice	45	160	Ford	Alice

The (unrestricted) cartesian product has *unbounded* sensitivity.

Join $\bowtie$

$$R \underset{R.a_i = T.a_i}{\bowtie} T = \sigma_{(R.a_i = T.a_i)}(R \times T)$$

Intermediate query sensitivity

- op any of $\cup, \cap, \setminus, \sigma, \pi, \times, \times_1$
- $\text{op} : (\mathcal{R}^n, d_{nH}) \to (\mathcal{R}, d_H)$
- C_{op} constraint obtained *after* op

$$
\begin{array}{lcll}
S(Id) & = & 1 & \text{base case} \\
S(\text{op} \circ Q) & = & \Delta_{\text{op}} \cdot S(Q) & \text{if } n = 1 \\
S(\text{op} \circ (Q_1, Q_2)) & = & \Delta_{\text{op}} \cdot \max(S(Q_1), S(Q_2)) & \text{if } n = 2
\end{array}
$$

Intermediate query sensitivity

- op any of $\cup, \cap, \setminus, \sigma, \pi, \times, \times_1$
- $\text{op} : (\mathcal{R}^n, d_{nH}) \to (\mathcal{R}, d_H)$
- C_{op} constraint obtained *after* op

$$
\begin{array}{lll}
S(Id) & = \min(1, diam(C_{Id})) & \text{base case} \\
S(\text{op} \circ Q) & = \min\left(\Delta_{\text{op}} \cdot S(Q), diam(C_{\text{op} \circ Q})\right) & \text{if } n = 1 \\
S(\text{op} \circ (Q_1, Q_2)) & = \min\left(\Delta_{\text{op}} \cdot \max(S(Q_1), S(Q_2)), diam(C_{\text{op} \circ (Q_1, Q_2)})\right) & \text{if } n = 2
\end{array}
$$

Aggregation γ

$$_{\{a_1,\ldots,a_m\}}\gamma_{\{f_1,\ldots,f_k\}} : (\mathcal{R}, d_H) \to (\mathcal{R}, d_H)$$

- groups tuples with the same values of a_i
- computes f_j for each group (`count`,`max`,`min`,`avg`,`sum`)
- returns a single tuple for each group, with a_i and f_j.

```
SELECT   Car, Count(*), Avg(Height)
FROM     R
GROUPBY Car
```

$$_{\{Car\}}\gamma_{\{Count, Avg(Height)\}} \left(\begin{array}{llll} \text{Name} & \text{Age} & \text{Height} & \text{Car} \\ \hline \text{Alice} & 45 & 160 & \text{Ford} \\ \text{John} & 30 & 180 & \text{Fiat} \\ \text{Frank} & 45 & 165 & \text{Bmw} \\ \text{Eve} & 20 & 170 & \text{Ford} \end{array} \right) = \begin{array}{lll} \text{Car} & \text{Count} & \text{Avg(Height)} \\ \hline \text{Ford} & 2 & 165 \\ \text{Fiat} & 1 & 180 \\ \text{Bmw} & 1 & 165 \end{array}$$

Functions

Assume $\emptyset\gamma_f$

Functions

Assume ${}_{\emptyset}\gamma_{f}$

$$R \sim R'$$

$$\begin{aligned}
\Delta_{\texttt{count}}(C) &= 1 \\
\Delta_{\texttt{sum}_{a_i}}(C) &= \max\{|\sup(C, a_i)|, |\inf(C, a_i)|\} \\
\Delta_{\texttt{avg}_{a_i}}(C) &= |\sup(C, a_i) - \inf(C, a_i)| \times \tfrac{1}{2} \\
\\
\Delta_{\texttt{max}_{a_i}}(C) &= |\sup(C, a_i) - \inf(C, a_i)| \\
\Delta_{\texttt{min}_{a_i}}(C) &= |\sup(C, a_i) - \inf(C, a_i)|
\end{aligned}$$

Functions

Assume ${}_{\emptyset}\gamma_{f}$

$$d_H(R, R') = n$$

$$\begin{aligned}
\Delta_{\mathtt{count}}(C) &= 1 && \times n \\
\Delta_{\mathtt{sum}_{a_i}}(C) &= \max\{|\sup(C, a_i)|, |\inf(C, a_i)|\} && \times n \\
\Delta_{\mathtt{avg}_{a_i}}(C) &= |\sup(C, a_i) - \inf(C, a_i)| && \times \tfrac{n}{1+n} \\
\\
\Delta_{\mathtt{max}_{a_i}}(C) &= |\sup(C, a_i) - \inf(C, a_i)| \\
\Delta_{\mathtt{min}_{a_i}}(C) &= |\sup(C, a_i) - \inf(C, a_i)|
\end{aligned}$$

Global Sensitivity

Definition (global sensitivity)

The global sensitivity GS of a query $\gamma_f(Q)$ is defined as:

$$GS(\gamma_f(Q)) = \begin{cases} \Delta_f(C_Q) \cdot S(Q) & \text{if } f = \texttt{count}, \texttt{sum}, \texttt{avg} \\ \Delta_f(C_Q) & \text{if } f = \texttt{max}, \texttt{min} \end{cases}$$

Global Sensitivity

Definition (global sensitivity)

The global sensitivity GS of a query $\gamma_f(Q)$ is defined as:

$$GS(\gamma_f(Q)) \quad = \quad \begin{cases} \Delta_f(C_Q) \cdot S(Q) & \text{if } f = \texttt{count}, \texttt{sum}, \texttt{avg} \\ \Delta_f(C_Q) & \text{if } f = \texttt{max}, \texttt{min} \end{cases}$$

Theorem (Soundness and strictness)

The sensitivity bound computed by $GS(\cdot)$ is sound and strict. Namely:

$$GS(\gamma_f(Q)) \; = \; \Delta_{\gamma_f(Q)}$$

Example

c-schema $(\{Weight, Height\}, C_I)$

$$C_I = \{Weight \in [0, 150] \ \wedge \ Height \in [0, 200]\}$$

Example

c-schema $(\{Weight, Height\}, C_I)$

$$C_I = \{Weight \in [0, 150] \ \wedge \ Height \in [0, 200]\}$$

$$\gamma_{avg(Weight)}(\sigma_{Weight \leq Height - 100}(R))$$

Example

c-schema $(\{Weight, Height\}, C_I)$

$$C_I = \{Weight \in [0, 150] \ \wedge \ Height \in [0, 200]\}$$

$$\gamma_{avg(Weight)}(\sigma_{Weight \leq Height - 100}(R))$$

$$C_Q = \{Weight \in [0, 150] \ \wedge \ Height \in [0, 200] \ \wedge \ Weight \leq Height - 100\}$$

Example

c-schema $(\{Weight, Height\}, C_I)$

$$C_I = \{Weight \in [0, 150] \ \wedge \ Height \in [0, 200]\}$$

$$\gamma_{avg(Weight)}(\sigma_{Weight \leq Height - 100}(R))$$

$$C_Q = \{Weight \in [0, 150] \ \wedge \ Height \in [0, 200] \ \wedge \ Weight \leq Height - 100\}$$

$$\Delta(C_I, \gamma_{avg(Weight)}) = \frac{|max(C_I, Weight) - min(C_I, Weight)|}{2} = 75$$

$$\Delta(C_Q, \gamma_{avg(Weight)}) = \frac{|max(C_Q, Weight) - min(C_Q, Weight)|}{2} = 50$$

Future Work

Join datasets

Join $\bowtie$

- $\times_n$: product with blocks of a fixed n size , to obtain n sensitivity. policies to pick these representative elements
- $\times_\gamma$: single record is built as an aggregation of the relation, thus falling in the case of $\times_1$ sensitivity
- a mix the two approaches could be considered, building n aggregations, possibly using the operator ${}_{\{a_i\}}\gamma_f$

Compositional ϵ analysis

Sequential composition: Q_i queries each providing ϵ_i differential privacy, $Q_n \circ \ldots \circ Q_1$ provides $(\sum_i \epsilon_i)$-differential privacy.

Parallel composition: Q_i queries each providing ϵ differential privacy, parallel application to disjoint subsets of the input provides again ϵ-differential privacy.

Extend static analysis to compute ϵ and optimize to parallelize

Comparison of metric spaces

Distance d_{nH}

$$d_{nH}((R_1, \ldots, R_n), (R'_1, \ldots, R'_n)) = \max(d_H(R_1, R'_1), \ldots, d_H(R_n, R'_n))$$

At first was the Manhattan distance

$$d_{2H}((R_1, R_2), (R_3, R_4)) = d_H((R_1, R_2)) + d_H(R_3, R_4)$$

- lower sensibilities on many operators, namely $\cup, \cap, \setminus$ all had sensitivity 1
- did not allow us to compute the real sensitivity but just a bound

Explore further the comparison

Thanks