Inference Control for Statistical Databases

Philip W. L. Fong

Department of Computer Science
University of Calgary
Calgary, Alberta, Canada

CPSC 525/625 (Fall 2016)
Outline

1. Basic Definitions

2. Small and Large Query Set Attacks

3. Tracker Attacks
   - Individual Tracker
   - General Trackers

4. Enhancing Privacy in Databases
**Abstract View of a Statistical Database**

**FIGURE 6.1 Abstract view of a statistical database.**

<table>
<thead>
<tr>
<th>Record</th>
<th>$A_1 \ldots A_j \ldots A_M$</th>
<th>$x_{11} \ldots x_{1j} \ldots x_{1M}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>$x_{11} \ldots x_{1j} \ldots x_{1M}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.</td>
</tr>
<tr>
<td>$i$</td>
<td>$x_{i1} \ldots x_{ij} \ldots x_{iM}$</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.</td>
</tr>
<tr>
<td>$N$</td>
<td>$x_{N1} \ldots x_{Nj} \ldots x_{NM}$</td>
<td>.</td>
</tr>
</tbody>
</table>
### Example

**TABLE 6.1 Statistical database with \( N = 13 \) students.**

<table>
<thead>
<tr>
<th>Name</th>
<th>Sex</th>
<th>Major</th>
<th>Class</th>
<th>SAT</th>
<th>GP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allen</td>
<td>Female</td>
<td>CS</td>
<td>1980</td>
<td>600</td>
<td>3.4</td>
</tr>
<tr>
<td>Baker</td>
<td>Female</td>
<td>EE</td>
<td>1980</td>
<td>520</td>
<td>2.5</td>
</tr>
<tr>
<td>Cook</td>
<td>Male</td>
<td>EE</td>
<td>1978</td>
<td>630</td>
<td>3.5</td>
</tr>
<tr>
<td>Davis</td>
<td>Female</td>
<td>CS</td>
<td>1978</td>
<td>800</td>
<td>4.0</td>
</tr>
<tr>
<td>Evans</td>
<td>Male</td>
<td>Bio</td>
<td>1979</td>
<td>500</td>
<td>2.2</td>
</tr>
<tr>
<td>Frank</td>
<td>Male</td>
<td>EE</td>
<td>1981</td>
<td>580</td>
<td>3.0</td>
</tr>
<tr>
<td>Good</td>
<td>Male</td>
<td>CS</td>
<td>1978</td>
<td>700</td>
<td>3.8</td>
</tr>
<tr>
<td>Hall</td>
<td>Female</td>
<td>Psy</td>
<td>1979</td>
<td>580</td>
<td>2.8</td>
</tr>
<tr>
<td>Iles</td>
<td>Male</td>
<td>CS</td>
<td>1981</td>
<td>600</td>
<td>3.2</td>
</tr>
<tr>
<td>Jones</td>
<td>Female</td>
<td>Bio</td>
<td>1979</td>
<td>750</td>
<td>3.8</td>
</tr>
<tr>
<td>Kline</td>
<td>Female</td>
<td>Psy</td>
<td>1981</td>
<td>500</td>
<td>2.5</td>
</tr>
<tr>
<td>Lane</td>
<td>Male</td>
<td>EE</td>
<td>1978</td>
<td>600</td>
<td>3.0</td>
</tr>
<tr>
<td>Moore</td>
<td>Male</td>
<td>CS</td>
<td>1979</td>
<td>650</td>
<td>3.5</td>
</tr>
</tbody>
</table>
Selecting a Subset of Tuples

- **Characteristic formula \( C \):**
  analogous to the **WHERE** clause in the **SELECT** command of SQL.
Selecting a Subset of Tuples

- **Characteristic formula $C$:** analogous to the `WHERE` clause in the `SELECT` command of SQL.
- **Example:**

  $\left( \text{Sex} = \text{Male} \right) \cdot \left( \left( \text{Major} = \text{CS} \right) + \left( \text{Major} = \text{EE} \right) \right)$
Selecting a Subset of Tuples

- **Characteristic formula C:** analogous to the **WHERE** clause in the **SELECT** command of **SQL**.
  
  Example:

  \[(Sex = Male) \cdot ((Major = CS) + (Major = EE))\]

- **Meaning of connectives:**
  - and  \(\land\)  \(\cdot\)
  - or  \(\lor\)  \(+\)
  - not  \(\neg\)  \(~\)
Selecting a Subset of Tuples

- **Characteristic formula** $C$: analogous to the `WHERE` clause in the `SELECT` command of SQL.

  **Example:**

  $$(Sex = Male) \cdot ((Major = CS) + (Major = EE))$$

- **Meaning of connectives:**
  - and $\land \bullet$
  - or $\lor +$
  - not $\neg \sim$

- **Shorthand:**
  $$(Male) \cdot ((CS) + (EE))$$
Selecting a Subset of Tuples

- **Characteristic formula C**: analogous to the `WHERE` clause in the `SELECT` command of SQL.

- Example:

  \[(Sex = Male) \bullet ((Major = CS) + (Major = EE))\]

- Meaning of connectives:
  - and `\&` `\cdot`
  - or `\lor` `\lor`
  - not `\neg` `\sim`

- Shorthand:

  \[(Male) \bullet ((CS) + (EE))\]

- Overload C to refer to both the characteristic formula as well as the subset of tuples (aka query set) defined by C.
**Statistical Queries**

- \( \text{count}(C) = |C| \)

In the following, when we write \( q(C) \) we mean one of \( \text{count}(C) \) or \( \sum (C, A_j) = \Sigma_{i \in C} x_{ij} \). We focus on \( \text{count} \) and \( \text{sum} \) as other statistics can be derived from them.
Statistical Queries

- **count**$(C) = |C|$
- **sum**$(C, A_j) = \Sigma_{i \in C} x_{ij}$
Statistical Queries

- **count**($C$) = $|C|$
- **sum**($C$, $A_j$) = $\sum_{i \in C} x_{ij}$
- We focus on **count** and **sum** as other statistics can be derived from them.

$$\text{avg}(C, A_j) = \frac{\text{sum}(C, A_j)}{\text{count}(C)}$$
**Statistical Queries**

- \( \text{count}(C) = |C| \)
- \( \text{sum}(C, A_j) = \sum_{i \in C} x_{ij} \)
- We focus on \( \text{count} \) and \( \text{sum} \) as other statistics can be derived from them.

\[
\text{avg}(C, A_j) = \frac{\text{sum}(C, A_j)}{\text{count}(C)}
\]

- In the following, when we write \( q(C) \) we mean one of \( \text{count}(C) \) or \( \text{sum}(C, A_j) \).
Outline

1. Basic Definitions
2. Small and Large Query Set Attacks
3. Tracker Attacks
   - Individual Tracker
   - General Trackers
4. Enhancing Privacy in Databases
Suppose the following two queries are issued in sequence:

- \( \text{sum}(EE \cdot Female) = 1 \)
- \( \text{sum}(EE \cdot Female, GP) = 2.5 \)
### TABLE 6.1 Statistical database with \( N = 13 \) students.

<table>
<thead>
<tr>
<th>Name</th>
<th>Sex</th>
<th>Major</th>
<th>Class</th>
<th>SAT</th>
<th>GP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allen</td>
<td>Female</td>
<td>CS</td>
<td>1980</td>
<td>600</td>
<td>3.4</td>
</tr>
<tr>
<td>Baker</td>
<td>Female</td>
<td>EE</td>
<td>1980</td>
<td>520</td>
<td>2.5</td>
</tr>
<tr>
<td>Cook</td>
<td>Male</td>
<td>EE</td>
<td>1978</td>
<td>630</td>
<td>3.5</td>
</tr>
<tr>
<td>Davis</td>
<td>Female</td>
<td>CS</td>
<td>1978</td>
<td>800</td>
<td>4.0</td>
</tr>
<tr>
<td>Evans</td>
<td>Male</td>
<td>Bio</td>
<td>1979</td>
<td>500</td>
<td>2.2</td>
</tr>
<tr>
<td>Frank</td>
<td>Male</td>
<td>EE</td>
<td>1981</td>
<td>580</td>
<td>3.0</td>
</tr>
<tr>
<td>Good</td>
<td>Male</td>
<td>CS</td>
<td>1978</td>
<td>700</td>
<td>3.8</td>
</tr>
<tr>
<td>Hall</td>
<td>Female</td>
<td>Psy</td>
<td>1979</td>
<td>580</td>
<td>2.8</td>
</tr>
<tr>
<td>Iles</td>
<td>Male</td>
<td>CS</td>
<td>1981</td>
<td>600</td>
<td>3.2</td>
</tr>
<tr>
<td>Jones</td>
<td>Female</td>
<td>Bio</td>
<td>1979</td>
<td>750</td>
<td>3.8</td>
</tr>
<tr>
<td>Kline</td>
<td>Female</td>
<td>Psy</td>
<td>1981</td>
<td>500</td>
<td>2.5</td>
</tr>
<tr>
<td>Lane</td>
<td>Male</td>
<td>EE</td>
<td>1978</td>
<td>600</td>
<td>3.0</td>
</tr>
<tr>
<td>Moore</td>
<td>Male</td>
<td>CS</td>
<td>1979</td>
<td>650</td>
<td>3.5</td>
</tr>
</tbody>
</table>
Unique Identification

- $\text{count}(C) = 1$ reveals that $C$ uniquely identifies an individual $I$. 
Unique Identification

- \( \text{count}(C) = 1 \) reveals that \( C \) uniquely identifies an individual \( I \).

- Attack:

\[
\text{count}(C \cdot D) = \begin{cases} 
1 & \text{individual } I \text{ has } D \\
0 & \text{individual } I \text{ does not have } D
\end{cases}
\]
Small $C$

Suppose $I \in C$ but $\text{count}(C) > 1$. 
Suppose $I \in C$ but $\text{count}(C) > 1$.

If $\text{count}(C \cdot D) = \text{count}(C)$ then $I$ has $D$. 
• Suppose \( I \in C \) but \( \text{count}(C) > 1 \).

• If \( \text{count}(C \cdot D) = \text{count}(C) \) then \( I \) has \( D \).

• Otherwise, \( \text{count}(C \cdot D) < \text{count}(C) \), and we can’t infer anything about whether \( I \) has \( D \).
Suppose \( I \in C \) but \( \text{count}(C) > 1 \).

If \( \text{count}(C \cdot D) = \text{count}(C) \) then \( I \) has \( D \).

Otherwise, \( \text{count}(C \cdot D) < \text{count}(C) \), and we can’t infer anything about whether \( I \) has \( D \).

One-sided inference still possible!
Large $C$

- Suppose $I \in C$ and $\text{count}(C) = 1$
Suppose \( I \in C \) and \( \text{count}(C) = 1 \)

Large query sets also disclose information:

\[
\text{count}(\sim(C \cdot D)) = \begin{cases} 
N & \text{if } I \text{ does not have } D \\
N - 1 & \text{if } I \text{ has } D 
\end{cases}
\]
Suppose \( I \in C \) and \( \text{count}(C) = 1 \)

Large query sets also disclose information:

\[
\text{count}(\sim(C \cdot D)) = \begin{cases} 
N & I \text{ does not have } D \\
N - 1 & I \text{ has } D
\end{cases}
\]

Can also discover the exact value of attributes:

\[
\text{sum}(C, A) = \text{sum}(\text{All}, A) - \text{sum}(\sim C, A)
\]
• Suppose \( I \in C \) and \( \text{count}(C) = 1 \)

• Large query sets also disclose information:

\[
\text{count}(\sim(C \cdot D)) = \begin{cases} 
N & \text{I does not have } D \\
N - 1 & \text{I has } D 
\end{cases}
\]

• Can also discover the exact value of attributes:

\[
\text{sum}(C, A) = \text{sum}(All, A) - \text{sum}(\sim C, A)
\]

• In general,

\[
q(C) = q(All) - q(\sim C)
\]
A statistic $q(C)$ is permitted only if

$$n \leq |C| \leq N - n$$

where $n \geq 0$ is a parameter of the database.

**NB:** $n \leq N/2$ if any statistics at all are to be released.
Outline

1. Basic Definitions

2. Small and Large Query Set Attacks

3. Tracker Attacks
   - Individual Tracker
   - General Trackers

4. Enhancing Privacy in Databases
Suppose individual $I$ is uniquely identified by $C$. 
Suppose individual $I$ is uniquely identified by $C$.

But query-set-size control does not permit an attacker to test $\text{count}(C) \neq \text{count}(C \cdot D)$.
Suppose individual $I$ is uniquely identified by $C$.

But query-set-size control does not permit an attacker to test $\text{count}(C) \overset{?}{=} \text{count}(C \bullet D)$

What else can the attacker do?
Suppose $C$ can be “decomposed” as follows:

\[ C = C_1 \bullet C_2 \]

such that

\[ n \leq \text{count}(C_1 \bullet \sim C_2) \leq \text{count}(C_1) \leq N - n \]
Suppose $C$ can be “decomposed” as follows:

$$C = C_1 \cdot C_2$$

such that

$$n \leq \text{count}(C_1 \cdot \sim C_2) \leq \text{count}(C_1) \leq N - n$$

Now both $C_1$ and $C_1 \cdot \sim C_2$ are permitted by query-set-size control.
Suppose $C$ can be “decomposed” as follows:

$$C = C_1 \cdot C_2$$

such that

$$n \leq \text{count}(C_1 \cdot \sim C_2) \leq \text{count}(C_1) \leq N - n$$

Now both $C_1$ and $C_1 \cdot \sim C_2$ are permitted by query-set-size control.

Then the pair $(C_1, C_1 \cdot \sim)$ is called the individual tracker of $I$. 
Individual Tracker Attack (1)

- Let $T = C_1 \sim C_2$
- Recovering $\text{count}(C)$:
  
  $$\text{count}(C) = \text{count}(C_1) - \text{count}(T)$$

- More generally,

  $$q(C) = q(C_1) - q(T)$$

- If $\text{count}(C) = 1$, then the attacker can deduce the value of $A$ for $I$

  $$\text{sum}(C, A) = \text{sum}(C_1, A) - \text{sum}(T, A)$$
Illustration of Individual Tracker Attack (1)

\[ a) \ q(C) = q(C_1) - q(T) \]

\[ C = C_1 \quad \bullet \quad C_2 \]
\[ T = C_1 \quad \bullet \quad \sim C_2 \]
Again, let \( T = C_1 \cdot \sim C_2 \).

Recovering \( \text{count}(C \cdot D) \)

\[
\text{count}(C \cdot D) = \text{count}(T + C_1 \cdot D) - \text{count}(T)
\]

- If \( \text{count}(C \cdot D) = 0 \) then \( I \) does not have \( D \).
- If \( \text{count}(C \cdot D) = \text{count}(C) \) then \( I \) has \( D \).
Again, let $T = C_1 \cdot \sim C_2$.

- Recovering $\text{count}(C \cdot D)$

\[
\text{count}(C \cdot D) = \text{count}(T + C_1 \cdot D) - \text{count}(T)
\]

- If $\text{count}(C \cdot D) = 0$ then I does not have $D$.
- If $\text{count}(C \cdot D) = \text{count}(C)$ then I has $D$.

- Why is $\text{count}(T + C_1 \cdot D)$ permitted?
Illustration of Individual Tracker Attack (2)

\[ q(C \cdot D) = q(T + C1 \cdot D) - q(T) \]
Even if $C_1$ and $T$ are not in the range $[n, N - n]$, they can be replaced with permitted set $C_1 + C_M$ and $T + C_M$, where

$$\text{count}(C_1 \cdot C_M) = 0$$

$C_M$ is called the “mask”
- pad the small query sets with enough irrelevant records to put them in the permitted range
Even if $C_1$ and $T$ are not in the range $[n, N - n]$, they can be replaced with permitted set $C_1 + C_M$ and $T + C_M$, where

$$\text{count}(C_1 \bullet C_M) = 0$$

$C_M$ is called the "mask"
- pad the small query sets with enough irrelevant records to put them in the permitted range
- Why does this work?
General Tracker Motivation

- With individual tracker, a new tracker must be found for each person \((C = C_1 \cdot C_2)\).
- A single general tracker can be used to compute the answer to every restricted statistic in the database.
A general tracker is any characteristic formula $T$ such that

$$2n \leq |T| \leq N - 2n$$
A **general tracker** is any characteristic formula $T$ such that

$$2n \leq |T| \leq N - 2n$$

$q(T)$ is always permitted.
A general tracker is any characteristic formula $T$ such that

$$2n \leq |T| \leq N - 2n$$

$q(T)$ is always permitted.

In order for $T$ to exist, $n \leq N/4$
General Tracker Attack

1. Given: general tracker $T$, and restricted query $q(C)$
2. Compute $q(\text{All}) = q(T) + q(\sim T)$
3. If $|C| < n$ then compute:
   \[
   q(C) = q(C + T) + q(C + \sim T) - q(\text{All})
   \]
4. If $|C| > N - n$ then compute:
   \[
   q(C) = 2q(\text{All}) - q(\sim C + T) - q(\sim C + \sim T)
   \]
Illustration of General Tracker Attack

\[
\begin{array}{cc}
T & \sim T \\
C & w & x \\
\sim C & y & z \\
All & & &
\end{array}
\]

\[
q(All) = q(T) + q(\sim T) = w + x + y + z
\]

\[
q(C) = q(C + T) + q(C + \sim T) - q(All)
\]

\[
= (w + x + y) + (w + x + z) - (w + x + y + z)
\]

\[
= w + x
\]
Why does the following work?

\[ q(C) = 2q(\text{All}) - q(\sim C + T) - q(\sim C + \sim T) \]
General Tracker Attack

- Given: general tracker $T$, and restricted query $q(C)$
- Compute $q(\text{All}) = q(T) + q(\sim T)$
- If $|C| < n$ then compute:
  $$q(C) = q(C + T) + q(C + \sim T) - q(\text{All})$$
- If $|C| > N - n$ then compute:
  $$q(C) = 2q(\text{All}) - q(\sim C + T) - q(\sim C + \sim T)$$

- How do you know the above queries are permitted?
Finding a General Tracker

There are efficient algorithms for formulating a general tracker $T$. 
Outline

1. Basic Definitions
2. Small and Large Query Set Attacks
3. Tracker Attacks
   - Individual Tracker
   - General Trackers
4. Enhancing Privacy in Databases
Privacy-Enhancing Transformations

- **Suppression**: completely removing contents from certain entries
  - replace outliers of the age field by `null`
Privacy-Enhancing Transformations

- **Suppression**: completely removing contents from certain entries
  - replace outliers of the age field by `null`

- **Generalization**: replacing contents with an abstract version that convey less information
  - replacing age 24 by age range [20 – 25]
Privacy-Enhancing Transformations

- **Suppression**: completely removing contents from certain entries
  - replace outliers of the age field by `null`
- **Generalization**: replacing contents with an abstract version that convey less information
  - replacing age 24 by age range \([20 \rightarrow 25]\)
- **Noisification**: injecting noise into contents
  - add/subtract a randomly generated quantity to the age field of every record
  - age-related statistics are still correct with a predictable error
Privacy-Enhancing Transformations

- **Suppression**: completely removing contents from certain entries
  - replace outliers of the age field by `null`

- **Generalization**: replacing contents with an abstract version that convey less information
  - replacing age 24 by age range `[20 − 25]`

- **Noisification**: injecting noise into contents
  - add/subtract a randomly generated quantity to the age field of every record
  - age-related statistics are still correct with a predictable error

- **Permutation**: swapping entries
  - swapping the age field of records
  - age-related statistics are still correct.
Challenges

- What is hard is not the lack of methods to perturb a database.
- The real challenge lies in understanding when such perturbations can guarantee a notion of privacy.
Most of the materials in these slides are based on:
  - [Denning], Chapter 6
  - [Gollmann], Chapter 9