Implementation of Privacy Preservation of N-D Algorithms for Online Analytical Processing

Rohit Goel¹, Mahesh Kumar²
M.Tech Student, Dept. of CSE, Monad University, Hapur, India
Assistant Professor, Dept. of CSE, Monad University, Hapur, India

ABSTRACT: Online analytical processing (OLAP) is one of the most popular decision support and knowledge discovery techniques in business-intelligence systems. There are issues related to the protection of private information in Online Analytical Processing (OLAP) systems, where a major privacy concern is the adversarial inference of private information from OLAP query answers. This inference problem cannot be fully addressed by access control and data sanitization techniques.

KEYWORDS: preservation, analytical, problem, privacy.

1. INTRODUCTION

In computing, online analytical processing, or OLAP is one of the most popular decision support and knowledge discovery techniques in business-intelligence systems. It is an approach to swiftly answer multi-dimensional analytical queries. OLAP is part of the broader category of business intelligence, which also encompasses relational reporting and data mining.

We address issues related to the protection of private information in Online Analytical Processing (OLAP) systems, where a major privacy concern is the adversarial inference of private information from OLAP query answers. Most previous work on privacy preserving OLAP focuses on a single aggregate function and/or addresses only exact disclosure, which eliminates from consideration an important class of privacy breaches where partial information, but not exact values, of private data is disclosed (i.e., partial disclosure). We address privacy protection against both exact and partial disclosure in OLAP systems with mixed aggregate functions.

Typical applications of OLAP include business reporting for sales, marketing, management reporting, business process management, budgeting and forecasting, financial reporting and similar areas, with new applications coming up, such as agriculture. The term OLAP was created as a slight modification of the traditional database term OLTP (Online Transaction Processing).

In particular, we propose an information-theoretic inference control approach that supports a combination of common according to its communication aggregate functions (e.g., COUNT, SUM, MIN, MAX, and MEDIAN) and guarantees the level of privacy disclosure not to exceed thresholds predetermined by the data owners. We demonstrate that our approach is efficient and can be implemented in existing OLAP systems with little modification. It also satisfies the simulatable auditing model and leaks no private information through query rejections.

Through performance analysis, we show that compared with previous approaches, our approach provides more effective privacy protection while maintaining a higher level of query-answer availability.
II. WHY PRIVACY PRESERVATION IS REQUIRED FOR OLAP?

The data warehouse server holds private data, and is supposed to answer OLAP queries issued by users on the multidimensional aggregates of private data. However, it is a challenge to enable OLAP on private data without violating the data owners’ privacy.

A user may not have the right to access all individual data points in the data warehouse, but might be allowed to issue OLAP queries on the aggregates of data for which it has no right to access.

In an OLAP system, a privacy breach occurs if a user can infer certain information about a private data point for which it has no right to access from the query answers it receives as well as the data that it has the right to access. Such privacy breach is referred to as the **inference problem**.

**Example:**

In a hospital system, the accounting department (as a user) can access each patient’s financial data, but not the patients’ medical records. Nonetheless, the accounting department may query aggregate information related to the medical records, such as the total expense for patients with Alzheimer’s disease.

III. INFERENCE PROBLEM

In an OLAP system, a **privacy breach** occurs if a user can infer certain information about a private data point for which it has no right to access from the query answers it receives as well as the data that it has the right to access.

Such privacy breach is referred to as the **inference problem**. If owner of the cube makes Attribute Y of collection 1 as sensitive cell, then users do not have access to that sensitive cell, but they can request aggregate queries.

Suppose a user asks 2 queries:

1. What is the total no. of items in collection1?
2. What is the value of attribute X in collection 1?

**Answer 1. 47**
**Answer 2. 7**

**Inference:** The no. of items of Y in collection 1 is 47-7=40.

<table>
<thead>
<tr>
<th></th>
<th>April</th>
<th>May</th>
<th>June</th>
<th>July</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
<td>10</td>
<td>12</td>
<td>15</td>
<td>7</td>
<td>q5=47</td>
</tr>
<tr>
<td>CD</td>
<td>20</td>
<td>23</td>
<td>27</td>
<td>N/A</td>
<td>q6=70</td>
</tr>
<tr>
<td>DVD</td>
<td>23</td>
<td>35</td>
<td>16</td>
<td>36</td>
<td>q7=110</td>
</tr>
<tr>
<td>Game</td>
<td>N/A</td>
<td>25</td>
<td>30</td>
<td>14</td>
<td>q8=69</td>
</tr>
<tr>
<td>Sum</td>
<td>53</td>
<td>95</td>
<td>88</td>
<td>57</td>
<td></td>
</tr>
</tbody>
</table>

**Example of Inference Problem**

So the user is able to get the value of sensitive cell for which it has no access by requesting aggregate queries on the data.

So “**Privacy Breaching**” as user should not get value of sensitive cell but here it can be inferred.
IV. OLAP SYSTEM MODEL

Consider OLAP as a system where the data warehouse server stores data in an n-dimensional data cube, in order to support aggregate queries on an attribute-of-interest over selected data. We refer to such attribute-of-interest as the measure attribute. Besides the measure attribute, there are n dimension attributes, each of which is represented by a dimension of the data cube.

Each (base) cell of the data cube is the value of the measure attribute for the corresponding value combination of dimension attributes.

<table>
<thead>
<tr>
<th></th>
<th>April</th>
<th>May</th>
<th>June</th>
<th>July</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
<td>10</td>
<td>Known</td>
<td>15</td>
<td>Known</td>
<td>q5 = 25</td>
</tr>
<tr>
<td>CD</td>
<td>20</td>
<td>Known</td>
<td>27</td>
<td>Known</td>
<td>q6 = 47</td>
</tr>
<tr>
<td>DVD</td>
<td>Known</td>
<td>35</td>
<td>16</td>
<td>36</td>
<td>q7 = 87</td>
</tr>
<tr>
<td>Game</td>
<td>Known</td>
<td>25</td>
<td>Known</td>
<td>14</td>
<td>q8 = 39</td>
</tr>
<tr>
<td>Sum</td>
<td>q1 = 30</td>
<td>q2 = 60</td>
<td>q3 = 58</td>
<td>q4 = 50</td>
<td></td>
</tr>
</tbody>
</table>

For example, the table shows a two-dimensional data cube with a measure attribute of sales and two dimension attributes of product and time.

Each cell of the data cube in Table is the sales amount (i.e., measure attribute) of a product (e.g., Book) in a month (e.g., April). As in real cases, some cells in the data cube can be missing or not applicable.

V. INFERENC CONTROL

There are two types of methods that have been proposed to prevent inference problems from happening in OLAP systems:

1. Inference Control
2. Input/output perturbation.

1. Inference control:

In the inference control approach, after receiving a query from a user, the data warehouse server determines whether answering the query may lead to an inference problem, and then either rejects the query or answers it precisely.

2. Input/output Perturbation:

The input/output perturbation approach either perturbs (input) data stored in the data warehouse server with random noise and answers every query with estimation, or adds random noise to the (output) query answers in order to preserve privacy. But for decision making precise or exact answer is required, therefore inference control is better way for privacy information.

VI. IMPLEMENTATION AND RESULTS

1. INFERENC CONTROL ALGORITHM (for n-d Arbitrary Distribution)

Require: h-dimensional query q on sub-cube (a₁… aₙ₋₁, ALL… ALL), l₀ =0.
1: {When a query q is received.}
2: if function of q is MIN-like then
3: \( l_0 \leftarrow f_{max}(q, |q|)/H(x) \).
4: else if function of \( q \) is SUM-like then
5: for \( i \leftarrow 1 \) to \( n - h \) do
6: \( \Theta_i \leftarrow (a_1, a_i, \text{ALL}, a_{i+1}, \ldots a_{n-h}, \text{ALL}, \ldots, \text{ALL}). \)
7: Find \( T_i \leftarrow \{ \Theta' \mid \Theta' \in O_i, \Theta' \text{ is subset of } \Theta_i \} \)
8: \( \mu \leftarrow \min\{\min(\mu_k(\Theta) \mid \Theta' \in T_i) \mid d_i + d_{n-h+1} + d_{n-h+2} + \ldots + d_n - |T_i|d_i / 2, d_{n-h+1} / 2, \ldots, d_n / 2) \} \)
9: \( t \leftarrow \max(\text{the maximum integer that satisfies } \sum_{\Theta' \in T_i} \sigma_k(\Theta')(1/h) (1/h) |\Theta' \cap T_i| \geq t (d_i - \mu) + \left( t - 1 \right) (\mu - 1)(d_1 + d_2 + \ldots + d_{n-h}) / 2, \ldots, d_n / 2) \}
10: \( n_0 \leftarrow |q| - r_k(q) - t \).
11: \( l_0 \leftarrow \max(l_0, (1 - l_p)x (f_{max}(q, n_0) - f_{min}(q, n_0 - 1))/H(x)) \)
12: end for
13: end if
14: if \( l_p + l_0 \geq \min(l_0, l(q)) \) then
15: return \( \emptyset \). \{Reject query \( q \}\)
16: else
17: if function of \( q \) is SUM-like then
18: \( \mu_k(\Theta) \leftarrow \mu(\Theta) \)
19: \( T \leftarrow \{ \Theta' \mid \Theta' \in O_k, \Theta' \subseteq \Theta \} \)
20: \( \sigma_k(\Theta) \leftarrow \sigma(\Theta) - \sum_{\Theta' \in T} \sigma_k(\Theta') \)
21: \( O_k \leftarrow O_k \cup (\Theta, \mu_k(\Theta), \sigma_k(\Theta)) \)
22: end if
23: \( l_k \leftarrow \min(l_k, l(q)) \).
24: \( l_p \leftarrow l_p + l_0 \).
25: return \( q \). \{Answer query \( q \) correctly\}
26: end if

2. IMPLEMENTATION

Let \( r_k(q) \) be the number of cells in \( q \) that belong to \( G_k \) (i.e., known by \( C_k \) as preknowledge).

\( l(q) \) is the number of all cells in \( q \).

Let \( Q_k^x \) be the set of SUM-like queries in the query history \( Q_k \), and \( |Q_k^x| \) be the number of queries in \( Q_k^x \).

Let \( \sigma_k \) be the number of preknown cells covered by at least one query in \( Q_k^x \).

That is, \( \sigma_k = |\{x \mid x \in G_k, 3q \in Q_k^x \text{ such that } x \in q\}| \).

Let \( \mu_k \) be the minimum number of sensitive cells covered by a SUM-like query in the query history:

\[ \mu_k = \min(k(q) - r_k(q)) \quad q \in Q_k^x \]

In the algorithm, we use \( l_p \) denote the current upper-bound estimate on \( l_{max}(Q_k) \).

When \( Q_k = \emptyset \), the initial values of the parameters are \( \mu_k = \infty \), \( \sigma_k = 0 \), and \( l_p = 0 \).

With the algorithm, when a new query \( q \) is received, the data warehouse server computes an upper-bound estimate on \( l_{max}(q \mid Q_k) \) as \( l_0 \), and answers the query if and only if \( l_p + l_0 \) is less than the owner-specified threshold \( l \).

If \( l_p + l_0 < l \) (i.e., query answer \( q \) can be issued to user \( C_k \)), then the data warehouse server updates the values of \( \mu_k, \sigma_k, |Q_k^x|, \) and \( l_p \) in Steps 12-17.
As we can see, only $l_p$ needs to be updated for MIN-like queries while all four parameters are updated for SUM-like ones.

In the starting we are using “Analytical Workspace Manager” to analyze the data cube and to map the dimensions of data cube on corresponding tables of database. Then, these tables are used in program for query evaluation and query answering.

Here, instead of maintaining $\mu_k$ and $\sigma_k$ values for each user, we are maintaining $\mu_k$ and $\sigma_k$ values for each sub cube of query history.

3. ASSUMPTIONS

1) The algorithm is for n-dimensions.
2) The query must be entered in a special manner
   a) After every word space has to be given.
   b) First word of the query belongs to the aggregate function
   c) All the words except the first one are names of dimensions.
   d) Each query gives the value of measure attribute. e.g.- sum april
3) Our implementation gives whether the query must be answered or not.

Modification made in the algorithm

In line 9 of algorithm B (query evaluation of SUM like queries) we have modified the evaluation of ‘t’ such that

\[ 9. t \leftarrow \text{the maximum integer that satisfies } \sum \{ \sigma_k(\Theta') (h-1) / h (1/h) | \Theta' \in T_i \} \geq t (d_1-\mu) + (t) (\mu-1) (d_2+\ldots +d_h - h (t) \text{ assuming } 0 (0) =1. \]

9.1. $t_{prev} = t$;
9.2. $t_m = \max(t_{prev}, 0)$;
where, $t_{prev}$ is $t$ as calculated in the paper. $t_m$ is modified $t$.

4. INITIAL SETUP AND RESULTS

1. Sample Database used to show results for Algorithm.
   [Fig.4.1: Database details]

2. It is maintaining 3 text files-
   a) “dimensions.txt”- It is a text file containing the names of tables (dimensions) to be accessed by our program.
   [Fig.4.2: Dimensions details]
3. Program Details:
a) As soon as the program starts, `db_data_retrival()` establishes connection with database and `get_queryinfo()` takes query from the user and classifies it into “MIN like” or “SUM like” query.
b) `transfer2()` and `transfer()` functions transfer corresponding $\mu_k, \sigma_k$ and $l_p$ values of each subcube and subcube history into related data structures.
c) Depending on type of query, `query_evaluate()` evaluates the query and answers it if $l_o + l_p < l$ where $l_o$ = information accessed by the query, $l_p$=information accessed earlier  $l$=level at which privacy breach takes place.

4. Query asked: sum april
   Output: Query is answered
b) Query asked: sum book
   Output: Query is answered

   ![Fig.4.6: Query Result](image)

   ![Fig.4.7: Query Result](image)

c) Query asked: sum cd
   Output: Query is answered.

   ![Fig.4.7: Query Result](image)

d) Query asked: sum june
   Output: Query is rejected

   ![Fig.4.8: Query Result](image)
5. Updated text files are 
   a) “Queryhistory.txt” containing the subcube of answered queries.
   b) “u_k_s_k.txt” containing the corresponding \( \mu_k, \sigma_k \) and \( l_p \) values of answered query’s subcube

V. CONCLUSION

For implementing Inference Control Algorithm (A Generic n-d Algorithm), we have used “Analytical Workspace Manager” for building, analysing and mapping values. We have used Oracle 11g for database and mapped values on database tables with the help of “Analytical Workspace Manager”.

REFERENCES

1. Nan Zhang and Wei Zhao, “Privacy-Preserving OLAP-Theoretic Approach”, IEEE Transactions on Knowledge and Data Engineering. VOL 23, NO 1, January 2011.

BIOGRAPHY

I am pursuing Master of Technology in the Computer Science Engineering from Monad University, Hapur (U.P), India. I have done Bachelor of Technology degree in 2011 from Babu Banarsi Das Institute Of Engineering and Technology, Jahangiranaad(BSR), India. My research interests are Databasewarehouse.

Mr. Mahesh Kumar is an Assistant Professor in Monad University, Hapur(U.P), India.