DARM: A Privacy-preserving Approach for Distributed Association Rules Mining on Horizontally-partitioned Data

Presenter: Gaby Dagher

Omar Abdel Wahab, Concordia University
Moulay Omar Hachami, Concordia University
Arslan Zaffari, Concordia University
Mery Vivas, Concordia University
Gaby G. Dagher, Concordia University
Outline

1. Introduction
2. Literature Review
3. Problem Definition
4. Proposed Solution
5. Performance Evaluation
6. Conclusions
Motivation:

- Rapid evolution of data collection and storage technologies

- Extracting knowledge and hidden patterns from stored data has become a major necessity for individuals, companies, and government agencies.

- Applying data mining techniques to extract information is considered a challenge when the data is distributed over multiple owners
  - Each data owner is concerned about the privacy of individuals in his data.
Introduction

Motivating Scenario

<table>
<thead>
<tr>
<th>Zip Code</th>
<th>Age</th>
<th>Income</th>
<th>Noisy Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>[18-40]</td>
<td>[40k-75k]</td>
<td>2</td>
</tr>
<tr>
<td>H</td>
<td>[18-40]</td>
<td>[75k-90k]</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Zip Code</th>
<th>Age</th>
<th>Income</th>
<th>Noisy Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>[18-40]</td>
<td>[40k-75k]</td>
<td>4</td>
</tr>
<tr>
<td>H</td>
<td>[40-55]</td>
<td>[75k-90k]</td>
<td>3</td>
</tr>
</tbody>
</table>

Data Provider 2
(Survey Company)

Master Miner

Data Provider 1
(Survey Company)

Query:
min_sup=60%,
min_conf=65%,
Age=[18,40], Income=ANY

Our contribution:
Differentially Private Association Rules
Age=[18,40] → Income=[40k-75k]

Master Consumer
(Marketing Company)
Introduction

Challenges:

- **Data Privacy**
  - One data provider should not learn sensitive information about the data of other providers.

- **Data Utility**
  - The generated rules should satisfy the data consumer’s request and needs.

- **Protection against Inference Attacks**
  - Prevent the data consumer from inferring sensitive information about the individuals involved in the database.
Introduction

Contributions

**Contribution #1:** Propose a comprehensive privacy-preserving approach for answering association rules queries in a distributed environment.

**Contribution #2:** Protect all providers against inference attacks from data consumers by guaranteeing that the returned association rules satisfy $\varepsilon$-differential privacy.

**Contribution #3:** Preserve the privacy of the mined data by preventing each data provider from learning sensitive information about other data providers during the mining process.

**Contribution #4:** Protect the confidentiality of the data consumer’s query against the data providers.

**Contribution #5:** We conduct performance evaluation on real-life data, and show that our approach is both scalable and efficient.
Association Rules Mining [1], [2], [3], [4], [5], [6], [7]:

**Summary:** Study the problem of mining association rules in distributed and parallel manners, where the data is partitioned across several nodes.

**Limitations:** these approaches were mostly interested in increasing the efficiency of the mining process, while ignoring the privacy concerns that may arise from building a global mining model.
Privacy in Distributed Mining Models [8], [9], [10], [11], [12]:

**Summary:** Consider the privacy concerns that may arise from mining the data globally.

**Limitations:** rely on encryption to achieve privacy between data providers. However, a recent study shows that most encryption schemes are insufficient to guarantee data privacy and confidentiality, as the protocol on which they are based, namely precise query protocol (PQP), is vulnerable to attribute values inference.
Privacy-preserving Data Mashup [13], [14], [15], [16]:

**Summary:** Preserve the privacy of the data in a data mashup scenario.

**Limitations:** In contrary to our model which considers *privacy-preserving data mining* (PPDM), these approaches are designed to support *privacy-preserving data publishing* (PPDP) since they assume that the data itself will be shared among the different parties.
Outline

1. Introduction
2. Literature Review
3. Problem Definition
4. Proposed Solution
5. Performance Evaluation
6. Conclusions
Problem Definition

System Inputs:

1. **Association Rules Queries**: To obtain the set of strong association rules $R$ from the distributed data, the data consumer submits a query request $q$ to the master miner in which he specifies the minimum support threshold $\gamma$, the minimum confidence threshold $\alpha$, and a set of predicates $P$.

2. **$\epsilon$-differentially Private Data**: We assume that the data is horizontally partitioned into sub-tables each of which is hosted by one data provider.
   - Each data provider owns the same type of attribute information on different set of individuals.
Problem Definition

• **Adversary Model**
  Semi-honest, where each party is expected to follow the protocol correctly; however, it is curious and might try to infer sensitive information about the other parties.

• **Problem Statement**
  Given relational data $D$ that is horizontally partitioned into $n$ partitions, the objective is to design a privacy-preserving model for answering association rules queries in a distributed environment. The model must achieve three objectives: (1) to prevent each data provider from learning sensitive information about other data providers during the mining process, (2) to protect all providers against inference attacks from the data consumers, and (3) to preserve the confidentiality of each data consumer’s query against the data providers.
Outline

1. Introduction
2. Literature Review
3. Problem Definition
4. Proposed Solution
5. Performance Evaluation
6. Conclusions
Proposed Solution

• **Step 1** - Data Anonymization

• **Step 2** - Frequent Itemsets Generation

• **Step 3** - Association Rules Generation
Proposed Solution

Step 1: Data Anonymization:

- In this step, the data providers use the \( \varepsilon \)-differential privacy algorithm, called **DiffGen**, to anonymize their data and provide protection against linkage and inference attacks.

- Using **DiffGen**, the data owner makes sure that the regenerated data table provides privacy guarantee while being insensitive to any specific record.

- The *data anonymization* process can be divided into three main parts:
  1. Selecting a candidate attribute for specialization
  2. Determining the split value parameter
  3. Publishing the noisy counts
Proposed Solution

Table 2: Original Table

<table>
<thead>
<tr>
<th>Job</th>
<th>Age</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teacher</td>
<td>25</td>
<td>Female</td>
</tr>
<tr>
<td>Lawyer</td>
<td>51</td>
<td>Male</td>
</tr>
<tr>
<td>Painter</td>
<td>48</td>
<td>Female</td>
</tr>
<tr>
<td>Singer</td>
<td>20</td>
<td>Female</td>
</tr>
<tr>
<td>Dancer</td>
<td>32</td>
<td>Male</td>
</tr>
<tr>
<td>Lawyer</td>
<td>45</td>
<td>Male</td>
</tr>
<tr>
<td>Writer</td>
<td>39</td>
<td>Female</td>
</tr>
<tr>
<td>Doctor</td>
<td>58</td>
<td>Female</td>
</tr>
</tbody>
</table>

Table 3: Root Partition

<table>
<thead>
<tr>
<th>Job</th>
<th>Age</th>
<th>Sex</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any Job</td>
<td>(20-58)</td>
<td>3M5F</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 4: First Partition

<table>
<thead>
<tr>
<th>Job</th>
<th>Age</th>
<th>Sex</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professional</td>
<td>(25-58)</td>
<td>2M2F</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 5: Second Partition

<table>
<thead>
<tr>
<th>Job</th>
<th>Age</th>
<th>Sex</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artist</td>
<td>(20-48)</td>
<td>1M3F</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 6: Leaf Partition

<table>
<thead>
<tr>
<th>Job</th>
<th>Age</th>
<th>Sex</th>
<th>NoisyCount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artist</td>
<td>(20-48)</td>
<td>F</td>
<td>3+1=4</td>
</tr>
<tr>
<td>Artist</td>
<td>(20-48)</td>
<td>M</td>
<td>1-1=0</td>
</tr>
</tbody>
</table>
Proposed Solution

**Step 2: Frequent Itemsets Generation:**

- The master miner receives the data consumer’s query

- The master miner requests the support counts of all the attributes the data consumer is interested in from the different data providers

- The master miner generates all the possible frequent itemsets of different lengths subject to the *minimum support threshold* \( \gamma \) specified in the query.
**Algorithm 1:** Frequent Itemsets Generation

1: **Input:** minimum support threshold $\gamma$
2: **Input:** set of attributes $A$
3: **Input:** total number of rows of all the databases $t$
4: **Input:** Candidate itemset $C_k$ of length $k$
5: **Output:** frequent itemsets $F_k$ of length $k$
6: **procedure** FREQ ITEMSET GENERATION

7: $C_k = \emptyset$
8: $F_1 = \{\text{frequent items}\}$
9: **for** ($k = 1; F_k \neq \emptyset; k++$) **do**
10: $C_{k+1} =$ candidates generated from $F_k$
11: request support count $s(C_{k+1})$ from all providers
12: calculate $Support(C_{k+1}) = s(C_{k+1})/t$
13: **if** $Support(C_{k+1}) \geq \gamma$
14: $F_k = F_k \cup C_{k+1}$
15: **end if**
16: **end for**
17: **return** $\bigcup_k F_k$
18: **end procedure**
Step 3 - Association Rules Generation:

- Now that the frequent itemsets are known, the master miner generates all the possible combinations of the k-length \((k > 1)\) frequent itemsets that may constitute association rules.

- The master miner then sends these combinations to the data providers which separately calculate and send back the support counts of these combinations.

- The master miner computes the confidence of each association rule based on the feedback from the data providers.
  - For each association rule, if its confidence exceeds the \textit{minimum confidence threshold} \(\alpha\) specified by the data consumer, then the rule is considered a \textit{useful} rule.

- Finally, the master miner returns to the data consumer the set of all \textit{useful} association rules.
Proposed Solution

Algorithm 2: Association Rules Generation

1: **Input:** frequent itemsets $F$ divided into $L_i$ and $R_i$
2: **Input:** set of left-hand-side frequent itemsets $L_i$
3: **Input:** set of right-hand-side frequent itemsets $R_i$
4: **Input:** minimum confidence threshold $\alpha$
5: **Output:** set of strong association rules $R$
6: **procedure** ASSOCIATION_RULES_GENERATION
7:    **for each** combination $c \in L_i \rightarrow R_i$ **do**
8:        request support count $s(c)$ from all providers
9:        calculate $Confidence(c) = s(c)/s(L_i)$
10:       **if** $Confidence(c) \geq \alpha$
11:          $R = R \cup c$
12:       **end if**
13:    **end for**
14: **return** $R$
15: **end procedure**
Outline

1. Introduction
2. Literature Review
3. Problem Definition
4. Proposed Solution
5. Performance Evaluation
6. Conclusions
Performance Evaluation

Efficiency

Figure 3: Efficiency w.r.t. the number of attributes per a query for frequent itemsets generation phase, and association rules mining phase.
Performance Evaluation

Scalability

Figure 4: Scalability of query processing w.r.t. the number of data records in the database for frequent itemsets generation phase, and association rules mining phase.
Performance Evaluation

Efficiency w.r.t. nSpecializations

Figure 5: Efficiency w.r.t to the number of specializations during the anonymization process using DiffGen algorithm.
In this paper, we propose a comprehensive privacy-preserving approach for answering association rules queries in a distributed environment, with the goal of preserving both data privacy and query confidentiality.

The proposed approach (1) protects all providers against inference attacks from data consumers by guaranteeing that the returned association rules to the data consumer satisfy $\varepsilon$-differential privacy, (2) preserves the privacy of the mined data by preventing each data provider from learning sensitive information about other data providers during the mining process, and (3) protects the confidentiality of the data consumer’s query against the data providers such that the master miner is able to mine the association rules without revealing the query to the data providers.


Thank You...😊