Performing Data Mining in (SRMS) Through Vertical Approach with Association Rules

Ambarish S. Durani¹, Vinay Kapse²

Abstract: This system technique is used for efficient data mining in SRMS (Student Records Management System) through vertical approach with association rules in distributed databases. The current leading technique is that of Kantarcioğlu and Clifton[1]. In this system I deal with two challenges or issues, one that computes the union of private subsets that each of the interacting users hold, and another that tests the inclusion of an element held by one user in a subset held by another. The existing system uses different techniques for data mining purpose like Apriori algorithm. The Fast Distributed Mining (FDM) algorithm of Cheung et al. [2], which is an unsecured distributed version of the Apriori algorithm. Proposed system offers enhanced privacy and data mining with respect to the Encryption techniques and Association rule with Fp-Growth Algorithm in private cloud (system contains different files of subjects with respect to their branches). Due to this above techniques the expected effect on this system is that, it is simpler and more efficient in terms of communication cost and combinational cost. Due to these techniques it will affect the parameter like time consumption for execution, length of the code is decrease, find the data fast, extracting hidden predictive information from large databases and the efficiency of this system is increased by the 20%.

Keywords: Data Mining; Vertical Approach; Association Rules

1. Introduction

In recent years the sizes of databases has increased rapidly. This has led to a growing interest in the development of tools capable in the automatic extraction of knowledge from data. The term Data Mining, or Knowledge Discovery in Databases, has been adopted for a field of research dealing with the automatic discovery of implicit information or knowledge within databases [2]. In Previous System, here the problem of data mining of association rules in partitioned databases. In that setting, there are several departments (or users) that hold homogeneous databases, i.e., databases that share the same schema but hold information on different entities. The goal is to find all association rules with support at least s and confidence at least c, for some given minimal support size s and confidence level c, that hold in the unified database, while minimizing the information disclosed about the private databases held by those users[1]. Consider the application SRMS in which the different databases are situated in each department, where the all data is stored semester wise. The main aim behind this system that the data should be stored in optimal (minimize) format with some secure techniques. In SRMS the data is used in large scale so, this propose system provide some technique for data mining with encryption /decryption techniques in private cloud. How SRMS worked?

2. Process of Execution

- First SRMS is one kind of web application which is used by a particular organization. Where the modules are Used by all staffs and admin.
- Second, store data in separate file respect to branch and Semester on cloud using encryption algorithm like AES.
- Third, collect data for ex. Admin wants data of semester 3 of CSE branch then collect the data through Association rule then id and name is taken from static Database and only marks are collect from different file which is present on cloud.
- Finally, shown the combine record of related semester and branch.

3. Review Literature

The existing system uses different techniques for data mining purpose like Apriori algorithm. The Fast Distributed Mining (FDM) algorithm of Cheung et al. [2], which is an unsecured distributed version of the Apriori algorithm.

Data mining is not particularly new statisticians have used similar manual approaches to review data and provide business evolutions for many years. Changes and update in data mining techniques, however, have enabled organizations to collect, monitor, analyze, and access data in new ways. The first change occurred in the area of basic data collection. Before companies uses the transition from paper-based records to computer-based systems, managers had to wait for staff to give records of pieces together to know how well the business was performing or how current performance compared with previous. As all companies started collecting and saving basic data in computers, they were able to start quick answering detailed easily.

Now a day’s a third party is exist to provide a service for commercial company. The users could submitted to him their inputs and he would perform the function evaluation and send to them the resulting output. In the absence of such a trusted third party, it is needed to devise techniques that the users can run on their own in order to arrive at the required output. The next aim is to secure the inputs of each user. if the both are combined together(data mining and Secure) the third party involvement is avoided. Yao was the first to propose a generic solution for this problem in the case of two players. [3].

In our problem, the inputs are the partial databases, and the required output is the list of association rules that hold in the cache memory with support and confidence s and c, respectively. They can be applied only to small inputs and...
functions which are realizable by simple circuits. In more complex settings, such as ours, other methods are required for carrying out this computation.

For example[11], the rule found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy burger. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection and bioinformatics. Three parallel algorithms for mining association rules [12], an important data mining problem is formulated in this paper. These algorithms have been designed to investigate and understand the performance implications of a spectrum of trade-offs between computation, communication, memory usage, synchronization, and the use of problem-specific information in parallel data mining [13].

For mining of data and encryption/decryption different techniques are available. Like for data mining K-means algorithms, Apriori Algorithm, Fast Distributed Mining and for encryption/Decryption RSA, DES etc [10]. This paper proposes the Fp growth mining with AES algorithm to provide mining and encryption. Figure shows the architecture scheme for SRMS.

![Architecture Scheme](image_url)

**Figure 3.1: Architecture Scheme**

4. Methodology

A. Mining Algorithm

Previously Apriori algorithm is used. It uses a generate and test approach generates candidate item sets and tests if they are frequent [4]. For SRMS Apriori algorithm is not beneficial, one disadvantage is overcome by FP Growth algorithm. FP-Growth allows frequent itemset discovery without candidate itemset generation. Two step approach:

1. Build a compact data structure called the FP-tree
2. Extracts frequent itemsets directly from the FP-tree.

Mining is preferably used for a large amount of data [8, 9] and related algorithms often require large data sets to create quality models [7]. The relationship between data mining and cloud is worth to discuss. Cloud providers use data mining to provide clients a better service [6].

B. AES algorithm

AES is asymmetric which is encrypted by different keys. Here in this paper the AES is used with different key length, different iteration and perform operation of different file size. The encryption is in fact not difficult to break if a dictionary of words with their expected frequencies is available [5] this will covered in Result analysis.

C. Cloud Interface

Cloud is used for only storage purpose, now a day there are two options available for storing data, first is server and second is cloud. For better security and larger space cloud is a better option. It is quite difficult to locate path of cloud where actual data is stored. In SAMS the files are encrypted using AES algorithm. The files are stored in “SAMS” domain in cloud. The union of record is performed by FP algorithm and data is return to the admin module. Cloud is also shows the message that how many space is used by user and show remaining space.

5. Result Analysis

The result analysis is performed by the calculating the communication cost and combinational cost, with the encryption and decryption time. The readings are calculated by using different key length, file size and iterations. All experiments were implemented in C# (.net 4) and were executed on an Intel(R) Core(TM) i3 personal computer with a 1.66GHz CPU, 8 GB of RAM, and the 32-bit operating system Windows 7 ultimate. Table 5.1 Shows the computational and combinational cost for 32 key length and different file size.
Table 5.1: Computational cost and Combinational cost with encryption and decryption time.

<table>
<thead>
<tr>
<th>sr no</th>
<th>key length</th>
<th>iterations</th>
<th>Encrytion time</th>
<th>Decryption time</th>
<th>Cloud Store Time</th>
<th>communication cost</th>
<th>total store time</th>
<th>cloud retrieval time</th>
<th>combinational cost</th>
<th>file transfer rate MB/SEC</th>
<th>support</th>
<th>confidence</th>
<th>tamir tassa</th>
<th>tamir tassa</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32</td>
<td>16</td>
<td>10.0023</td>
<td>18.0023</td>
<td>1049</td>
<td>1059.0616</td>
<td>6022.345</td>
<td>6521.373</td>
<td>3.21</td>
<td>1.30</td>
<td>20</td>
<td>200000</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>12</td>
<td>4.0002</td>
<td>17.0001</td>
<td>1026</td>
<td>1073.0614</td>
<td>6020.344</td>
<td>6531.379</td>
<td>3.21</td>
<td>1.20</td>
<td>10</td>
<td>80000</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>32</td>
<td>8</td>
<td>5.0008</td>
<td>18.0021</td>
<td>1103</td>
<td>1118.0614</td>
<td>6018.344</td>
<td>6504.372</td>
<td>3.21</td>
<td>1.30</td>
<td>20</td>
<td>100000</td>
<td>600</td>
<td>600</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>6</td>
<td>4.0003</td>
<td>17.0009</td>
<td>1080</td>
<td>1072.0594</td>
<td>6020.344</td>
<td>6508.372</td>
<td>3.21</td>
<td>4.20</td>
<td>80</td>
<td>120000</td>
<td>800</td>
<td>800</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>8</td>
<td>5.0006</td>
<td>18.0021</td>
<td>1103</td>
<td>1118.0614</td>
<td>6018.344</td>
<td>6504.372</td>
<td>3.21</td>
<td>1.50</td>
<td>10</td>
<td>140000</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>

Example 2:
1) Fixed key length 16 key length, File Size 3293,5680,10440 KB, Iterations 2,4,8,12,16.
Table 5.2: shows the Communication, Combinational cost with encryption and decryption time

<table>
<thead>
<tr>
<th>sr no</th>
<th>key length</th>
<th>iterations</th>
<th>Encrypt time</th>
<th>Decrypt time</th>
<th>Cloud Store Time</th>
<th>communication cost</th>
<th>cloud retrieval time</th>
<th>combinational cost</th>
<th>file transfer rate MB/SEC</th>
<th>support</th>
<th>confidence</th>
<th>tamir tassa</th>
<th>tamir tassa</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td>4.005</td>
<td>18.001</td>
<td>1034</td>
<td>0.006</td>
<td>0.010</td>
<td>0.001</td>
<td>0.007</td>
<td>5.177</td>
<td>1</td>
<td>20</td>
<td>300000</td>
<td>60</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>12.001</td>
<td>44.002</td>
<td>1041</td>
<td>0.048</td>
<td>0.071</td>
<td>0.010</td>
<td>0.007</td>
<td>5.177</td>
<td>31</td>
<td>20</td>
<td>800000</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>12.002</td>
<td>26.002</td>
<td>1034</td>
<td>0.050</td>
<td>0.071</td>
<td>0.010</td>
<td>0.007</td>
<td>5.177</td>
<td>3</td>
<td>20</td>
<td>1000000</td>
<td>500</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>12.002</td>
<td>26.002</td>
<td>1034</td>
<td>0.050</td>
<td>0.071</td>
<td>0.010</td>
<td>0.007</td>
<td>5.177</td>
<td>3</td>
<td>20</td>
<td>1000000</td>
<td>500</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>12.002</td>
<td>26.002</td>
<td>1034</td>
<td>0.050</td>
<td>0.071</td>
<td>0.010</td>
<td>0.007</td>
<td>5.177</td>
<td>3</td>
<td>20</td>
<td>1000000</td>
<td>500</td>
</tr>
</tbody>
</table>

Figure 5.5: Shows communication cost

Figure 5.7: Shows Encryption Time

Figure 5.6: Shows combinational cost

Figure 5.8: Shows Decryption Time

Example 3:
1) Fixed key length 8 key length ,File Size 3293, 5680, 10440 KB, Iterations 2,4,8,12,16
Table 5.3: Shows the Communication, Combinational cost with encryption and decryption time

<table>
<thead>
<tr>
<th>sr no</th>
<th>key length</th>
<th>iteration</th>
<th>Encrypt time</th>
<th>Decrypt time</th>
<th>Cloud store Time</th>
<th>Cloud retrieval Time</th>
<th>communication cost</th>
<th>combinational cost</th>
<th>file transfer rate MB/SEC</th>
<th>support</th>
<th>confidence</th>
<th>tamir tassa communication cost in millsec</th>
<th>tamir tassa (Transfer Rate)MB/SEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>16</td>
<td>17.0011</td>
<td>35.0019</td>
<td>1127</td>
<td>238249</td>
<td>1135.0647</td>
<td>6054.345</td>
<td>1.00</td>
<td>10</td>
<td>20</td>
<td>120000</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>12</td>
<td>8.0006</td>
<td>35.0019</td>
<td>1087</td>
<td>238249</td>
<td>1135.0647</td>
<td>6054.345</td>
<td>1.00</td>
<td>20</td>
<td>40</td>
<td>80000</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>8</td>
<td>12.0006</td>
<td>35.0019</td>
<td>1127</td>
<td>238249</td>
<td>1135.0647</td>
<td>6054.345</td>
<td>1.00</td>
<td>40</td>
<td>60</td>
<td>100000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>4</td>
<td>12.0001</td>
<td>35.0019</td>
<td>1127</td>
<td>238249</td>
<td>1135.0647</td>
<td>6054.345</td>
<td>1.00</td>
<td>60</td>
<td>80</td>
<td>200000</td>
<td>800</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>2</td>
<td>23.0011</td>
<td>35.0019</td>
<td>1127</td>
<td>238249</td>
<td>1135.0647</td>
<td>6054.345</td>
<td>1.00</td>
<td>80</td>
<td>100</td>
<td>120000</td>
<td>80000</td>
</tr>
</tbody>
</table>

Figure 5.9: Shows communication cost

Figure 5.10: Shows combinational cost

Figure 5.11: Shows Encryption Time

Figure 5.12: Decryption Time
6. Conclusion

From the above experimental setup, the communication cost and combinational cost is optimized in small data entries. To apply these techniques in larger databases, the Dynamic FP-Growth algorithm is beneficial for mining. It provides security with cloud storage and mining with FP Growth algorithm. This experimental setup is applied for small database file size and produced optimized results.

7. Future Work

For larger records, the dynamic FP-growth Algorithm is used for mining which overcomes the limitation of Apriori and FP-Growth algorithm.

References


Author Profile

Mr. Ambarish Durani is M Tech pursuing in Computer Science & Engineering From Vidharbha Institute of Technology, Nagpur. Nagpur University (M.H.), India. His area of interest is Data Mining.

Prof. Vinay Kapse is M.E. in WCC from G. H. Raisoni, Nagpur. Nagpur University, (M.H.), India and now he is working as Assistant Professor in CSE Department, Vidharbha Institute of Technology, Nagpur. His area of interest is Data Mining and Algorithm.