Prefix Span Algorithm with Pseudo-Projection technique for Privacy Preservation

Sankari M and Thirukumar K

PG Scholar, Department of Computer Science and Engineering, Assistant Professor (SG) Dr. Mahalingam College of Engineering and Technology, Pollachi, India.

Received 28 January 2017; Accepted 22 March 2017; Available online 28 April 2017

ABSTRACT

Data mining is used to discover interesting and useful knowledge from massive data. Privacy Preservation in data mining is required for exchanging confidential information. Utility Mining is used to identify the item sets which have high utility value. It is defined in two aspects, internal and external utility. Internal utility consists of number of items in a given dataset with its quantity value. External utility consists of profit value for each item. To further improve the algorithm efficiency in terms of running time and storage utilization Utility Mining is used which ensures the privacy of data. In the existing method, Fast Perturbation using tree structure and tables (FPUTT) for privacy preserving utility mining is used to improve the data security, but it requires more time to protect the privacy because it conducts the database scanning operation repeatedly until the important information is hidden. To overcome this problem, Prefix Span algorithm is proposed, which uses prefix-projection method to reduce the candidate generation. Database projection is considered as major cost of Prefix Span. Hence Prefix Span integrated with Pseudo-Projection technique is proposed which reduces the cost of projected database and improves the efficiency of the proposed method. The performance evaluation outcomes the proposed Prefix Span achieves the better result compared to FPUTT in terms of Hiding Failure (HF), Misses Cost (MC) and Database Modification Ratio (DMR).

KEYWORDS: Privacy Preserving Utility Mining (PPUM), Pseudo-Projection technique, Fast Perturbation Using Tree Structure and Tables (FPUTT)

INTRODUCTION

Data mining algorithm refers to mining knowledge from large amount of data. It is used to find the frequent item set and uses a bottom up approach where frequent subsets are lengthen at each time from the given data. Sequential Pattern mining uses data mining algorithm to discover unexpected, useful patterns from the database [1].

Utility Mining is defined in two different aspects such as external and internal utility. External utility represents the weight or profit of items and the internal utility defines the quantity of items in a transaction. The item is said to be high utility item set, if its utility value should be greater than a user specified threshold. In business environments, privacy breaches implies that the specific information the data holders want to hide is illegally discovered by data analysing methods. Privacy Preservation is to achieve the data mining goals without affecting the privacy of the individuals [2].
Privacy preserving data mining algorithm is used to modify the original data by reducing the utility value of high sensitive item set and stored in a perturbed database so that the data and knowledge remain secure even after the mining process. Hence, the privacy is preserved and sensitive information cannot be leaked. Sensitive information is considered as a necessary data that can causes privacy breaches of data holders. The task is to reduce the candidate generation and represent all high utility item sets with no fail [3].

Hence, the concept of Privacy Preserving Utility Mining (PPUM), is used which is a combination of both Utility Pattern Mining (UPM) and Privacy Preserving Data Mining (PPDM). PPUM can deal with more important sensitive information. PPUM methods have been applied to the database perturbation process for effectively hiding sensitive utility items by removing certain items from original databases [4].

In this paper, database projection based on the pattern-growth method is applied in Prefix Span algorithm for mining sequential patterns. It helps to reduce the processing time by reducing the cost of projection which forms the projected database in recursive method. This ultimately improves the algorithm efficiency by handling the projected database in main memory. Hence pseudo-projection technique is proposed along with prefix Span which reduces the size of projected database [5].

Related Works:
A. GSP, SPADE and Prefix Span:
   There is a comparison between three kinds of algorithm GSP (Generalized Sequential Pattern) SPADE (Sequential Pattern Discovery using Equivalent Class) and Prefix Span (Prefix-projected Sequential Pattern). GSP is the Apriori based Horizontal formatting method, SPADE is the Apriori based vertical formatting method and is an efficient Algorithm for mining Frequent Sequences. Prefix-SPAN is Projection-based pattern growth method. GSP discovers sequential pattern and scans the database repeatedly depending on the length of the longest frequent sequences in database. Candidate generation is required. SPADE algorithm is used for fast discovery of sequential pattern (faster than GSP). It uses Apriori based approach and vertical format databases. Candidate generation is required. Prefix Span explores prefix-projection in sequential pattern mining. It mines the complete set of patterns and reduces the efforts of candidate sequence generation. It substantially reduces the size of projected databases and leads to efficient processing. GSP, SPADE, Prefix Span were executed on the same dataset. Hence the total time, Frequent Sequences Count and Max Memory taken by Prefix Span is very less compared to other two algorithms [6].

B. Prefix Projected Pattern:
   A projection-based, sequential pattern-growth approach is used for efficient mining of sequential patterns. Prefix Span algorithm is used which offers ordered growth and reduce projected databases. The major cost of Prefix Span is the construction of projected databases. If the number and the size of projected databases can be reduced, the performance of sequential pattern mining can be further improved. Pseudo Projection is the only technique which reduces the number and size of projected databases [7].

C. Privacy Preserving Utility Mining:
   Privacy Preserving Utility Mining (PPUM) algorithms such as Hiding high utility item first algorithm (HHUIF) and Maximum sensitive item sets conflict first algorithm (MSICF) are used to modify the database transactions containing sensitive item set which reduces the utility value and preventing reconstructing of original database from the perturbed one. The algorithm will hide the sensitive item set hence unauthorized users cannot mine them from the perturbed database. The algorithm balances privacy and knowledge discovery in sharing data. The tree construction is used for storing the sensitive values and it needs to search the whole database for every scan [8].

D. Log normal distribution:
   Utility Mining consists of two different aspects such as internal and external utility. The internal utility represents the quantities of products and the external utility represents the profit value for each item. The unit profits for items in utility tables are generated by using a log-normal distribution function and profit values are assigned randomly between 1 and 10 [9].

E. A fast perturbation algorithm:
   A fast perturbation using tree structures and tables is used for preserving the privacy. The header table is used for maintaining the most frequent items as a sensitive item. The sensitive item set is stored in tree structure and its frequency count is sorted in descending order. The database needs to reduce the utility value in the tree until it reaches the threshold value. Hence, it is time consuming. To further improve the process candidate generation can be reduced in future process [10].
System Architecture:

From a given dataset [15], the profit value for the transaction is calculated using log normal distribution and the frequent item subset is formed. Most frequent items are considered as a sensitive values and it is stored in a sequential pattern. Prefix Span algorithm is used to construct the projected database. It is based on projection of data that is the unique property of the algorithm among the other pattern mining algorithm. The Prefix Span mainly constructs the projected database in recursive manner to process the sequential data. Here, Pseudo-Projection technique is proposed along with the Prefix Span which reduces the cost of projection when a projected database can be held in main memory [7]. The System Architecture of Proposed System is shown in Figure 1.

![System Architecture Diagram](image)

**Fig. 1:** Architecture of Proposed System

**Modules:**

**A. Utility Mining:**

The pre-processed data contains both frequent and infrequent items. On partitioning the pre-processed data, the unique item set is identified and frequent subset is formed. In order to compute the frequent item set, original database is processed as input and sensitive item sets are discovered. The utility value is calculated by the product of external utility and internal utility. Using the frequent subset, the sensitive item set is formed [11].

<table>
<thead>
<tr>
<th>TID</th>
<th>Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>(B,6) (C,4) (D,3) (E,7)</td>
</tr>
<tr>
<td>T2</td>
<td>(B,5) (C,3) (D,9)</td>
</tr>
<tr>
<td>T3</td>
<td>(A,1) (B,6) (C,2) (D,5) (E,3)</td>
</tr>
<tr>
<td>T4</td>
<td>(A,3) (C,10) (D,3)</td>
</tr>
</tbody>
</table>

**Table 1:** a) Transaction table

<table>
<thead>
<tr>
<th>External Utility table</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
</tbody>
</table>

b) External Utility table

Given a database D composed of multiple transactions Ts and D is denoted as D = \{T_1, T_2, ..., T_n\}. Let I = \{i_1, i_2, ..., i_m\} be a set of items where each T is a subset of I, T belongs to I. Each T has a unique ID to identify them, called TID.

i) The internal utility, In(i_p, T_q), indicates the quantity of item \(i_p\) in \(T_q\).

For example, in Table 1(a), In(B,T_1) = 6 and In(C,T_4) = 10.

ii) The external utility Ex(i_p), represents the profit of \(i_p\).

For example in Table 1(b), Ex(A) = 5 and Ex(C) = 1.
iii) The utility of item $i_p$ in $T_q$, $U(i_p,T_q)$, is represented as
\[ U(i_p,T_q) = \ln(i_p,T_q) \times \text{Ex}(i_p) \]

For example in Table 1,
\[ U(CD,T_3) = \ln(C,T_3) \times \text{Ex}(C) + \ln(D,T_3) \times \text{Ex}(D) = 10 \times 1 + 3 \times 6 = 28 \]

iv) The utility of an item set $X$ in all the transactions of $D$ is the sum of the utilities of $X$ in each transaction.

The utility of item set $X$, $U(X)$, is
\[ U(X) = \sum(U(X,T_q)) \]

In Table 1, $U(AC) = U(AC,T_3) + U(AC,T_4) = 32$

Utility mining is to find all the item sets whose Values are beyond a user specified threshold. An item set $X$ is a high utility item set if $u(X) \geq \epsilon$, where $\epsilon$ is the minimum utility threshold.

X = \begin{cases} 
\text{Sensitive,} & u(X) > \epsilon \\
\text{Insensitive,} & u(X) < \epsilon 
\end{cases}

If $\epsilon$ is set to be 40, \{AC\} is a low utility item set and \{ACD\} is a high utility item set.

B. Prefix Span:

The sensitive item sets are represented in a sequential manner and Prefix Span discovers all frequent sequential patterns occurring in a given sequence. The given sequence is scanned to get the items that occurred frequently. Using divide and conquer method the sequence is partitioned and projected databases are constructed using prefix value for each items. From the projected databases the sequential patterns are mined from the database to reduce the repetition of items [12].

**PrefixSpan Algorithm:**

Step 1: Identification of length-l sequential pattern. The given sequence $S$ is scanned to get item that occurred frequently in $S$. The number of time the items occurred is equal to length-l of that item. Length-l is given by notation $\langle \text{pattern} \rangle: \langle \text{count} \rangle$.

Step 2: Divide search space. The complete set of sequential patterns can be partitioned into the following six subsets according to the six prefixes

1) the ones having prefix (a)…
2) the ones having prefix (f).

Step 3: Find subsets of sequential patterns. It constructs the projected databases and using the projected database sequential patterns is mined. Local frequent sequences are taken in projected databases to expand the sequential patterns.

**Example: Sequence database:**

<table>
<thead>
<tr>
<th>Sequence_id</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(ab)(ac)d(cf)</td>
</tr>
<tr>
<td>2</td>
<td>(ad)c(bc)(ae)</td>
</tr>
<tr>
<td>3</td>
<td>(ef)(ab)(df)c</td>
</tr>
<tr>
<td>4</td>
<td>eg(ab)c(bc)</td>
</tr>
</tbody>
</table>

i) The number of times the item occurred in a database is given by $a=4,b=4,c=4,d=3,e=3,f=3$

ii) Projected database: considering prefix value for a above sequences for example

Prefix $f = \{(ab)(df)(cb),(c)(bc)\}$

Prefix $c = \{(ac)(d(cf)),((bc)(ae)),(b),(bc)\}$

iii) Sequential patterns: Considering repeated value in sequence

Sequential patterns for $f = \{(f),(fb),(fbc),(fc),(fcb)\}$

$c = \{(c),(ca),(cb),(cc)\}$

C. Pseudo-Projection technique:

A Pseudo-Projection technique reduces the cost of projection when a projected database can be held in main memory. In the example, Sequence $(a(abc)(ac)(d)(c)(f))$ has postfixes $((abc)(ac)(d)(c)(f))$ and $((_c)(ac)(d)(c)(f))$ as projections in $(a)$ and $(ab)$ projected databases, respectively. They are redundant pieces of sequences. If the sequence database or projected database can be held in main memory, such redundancy can be avoided by pseudo-projection.

Instead of constructing a physical projection by collecting all the postfixes, we can use pointers referring to the sequences in the database. Each projection consists of two pieces of information that is pointer to the sequence in database and offset of the postfix in the sequence.
Pseudo-projection avoids physically copying postfixes. Hence, it is efficient in terms of both running time and space. Based on this observation, Prefix Span uses Pseudo-Projection once the projected databases can be held in main memory [13].

RESULTS AND DISCUSSIONS

A. Dataset:
The dataset is collected from Frequent Itemset Mining Implementations Repository (http://fimi.ua.ac.be/). Size of the dataset is 3.8MB and the average transaction size is 10. The unit profits for items are generated by using log-normal distribution.

B. Performance Evaluation:
Prefix span algorithm is more efficient than the existing algorithm because it uses divide and conquer method, hence the sequential pattern set is partitioned. The projected databases are constructed and sequential patterns are mined from the databases. One technique which reduces the number and size of projected databases is Pseudo-projection. Hence, Prefix Span integrated with Pseudo-Projection is more efficient and compact such that memory space is reduced and total time to find out the frequent item set is reduced.

The metrics below are used to check whether the sensitive item sets are hidden well and estimate side effect caused by each perturbation operation.

Hiding Failure (HF):
It is the ratio of sensitive itemsets that are disclosed before and after the perturbation process.

\[
HF = \frac{\text{No of sensitive itemsets from PDB}}{\text{No of sensitive itemsets}}
\]

Misses Cost (MC):
It is the modified ratio of insensitive itemsets among the results of the utility pattern mining. MC shows how many items are newly added or lost from PDB compared to the mining result of DB.

\[
MC = \frac{\text{No of new itemsets} + \text{No of lost insensitive itemsets}}{\text{No of Insensitive itemsets (from DB)}}
\]

Database Modification Ratio (DMR)
It is the difference ratio for the total utility between the DB and PDB.

\[
DMR = \frac{\text{Total utility of DB} - \text{Total utility of PDB}}{\text{Total utility of DB}}
\]

![Fig. a: Hiding Failure](image-url)
Figure a, b and c represent the privacy measure metrics such as Hiding Failure, Misses Cost, and Database Modification Ratio.

If MC is 0%, the high utility item sets of PDB have no side effect for the perturbation process. If HF becomes 0%, the result becomes ideal. If DMR is high, the difference between DB and PDB is also heavy. By having these values, it could be easy to identify whether the privacy preservation is preserved for the sensitive item sets.

Figure d shows the running time comparison of both FPUTT and Prefix Span algorithm with respect to time and length of sensitive item set. The result shows that the proposed algorithm takes less time to run compared to FPUTT.

**Conclusion And Future Work:**

In this paper, Prefix Span algorithm is integrated with the Pseudo-Projection technique. Hence, the efficiency for privacy preservation and time taken to generate the frequent item sets are enhanced. The previous algorithm, Fast Perturbation using a Tree Structure and Tables (called FPUTT) which can conduct the perturbation process with many database scans until it reaches the threshold value hence it takes time. To reduce the database scanning time the proposed system provides a way by using Pseudo-Projection technique which avoids costly candidate generation and repeated frequent item set generation by reducing the projected databases.
This paper shows that, how to efficiently prevent the privacy breaches caused by knowledge discovered from business or market databases. Proposed approach to data privacy protection can be enhanced by integrating present approach with web mining.

REFERENCES

1. JianPei, Jiawei Han, Behzad Mortazavi-Asl, 2004. “Mining Sequential Patterns by Pattern Growth: The Prefix Span Approach,” IEEE Transactions on Knowledge and Data Engineering, 16.
4. Li Liu, Murat Kantarcioglu, “The Application of the Perturbation Model-based Privacy Preserving Data Mining for Real-world Data,” Sixth IEEE International Conference on Data Mining Workshops (ICDMW’06).
10. Jia Wei Han, JianPei, YiwenYin, Runyiing Mao, 2004. “Mining Frequent Patterns without Candidate Generation:A Frequent Pattern Tree Approach,”Data Mining and Knowledge Discovery, 8: 53-87.