Efficient Mining and Hiding of Sensitive Association Rule

Sonali Sambhaji Shintre[1] & Prof. Pravin P. Kalyankar[2]

***Abstract*-There are several mining algorithms of association rules. One of the most popular algorithm is Apriori algorithm that is used to extract frequent itemsets from large databases. Based on this algorithm, this paper is organized into two parts in first part of paper an Improved Apriori Algorithm is being presented that efficiently generates association rules. These reduces unnecessary database scan at time of forming frequent large itemsets. In second part of this paper we have tried to give contribution to improved Apriori algorithm by hiding sensitive association rules which are generated by applying improved Apriori algorithm on supermarket database. In this paper we have used innovative approach that strategically modifies few transactions in transaction database to decrease support and confidence of sensitive rule without producing any side effects.**

***Index Terms*—Association rules, confidence, Data mining methods and Algorithm, Minimum Support Threshold (MST), Minimum Confidence Threshold (MCT), Rule hiding.**

I. INTRODUCTION

Data Mining is the process of discovering new patterns from large data sets. It extract useful information or knowledge from large database. Data mining has developed an important technology for large database. Data mining applications like business, marketing, medical analysis, products control and scientific etc[15], [16]. Association rule mining is one of the important problems in the data mining domain. Association rule analysis is a popular tool for discovering useful association from large database. This paper focuses on the problem of association rule mining, which hides the association rules from the database. This paper reviews the major method of rule hiding. Association rule mining, as a very important technique, has already been applied in a wide range of areas. As a result of association rule mining, many useful association rules will be discovered. To solve this, limit the mining process, in order to keep these sensitive rules being hidden. There are so many methods to solve this problem. In this paper we used Algorithm for this project is Improved Apriori Algorithm. Apriori is a classic algorithm for learning association rule. Apriori is designed to operate on databases containing transactions. Association rule is mainly based on discovering frequent item sets. Association rules are frequently used by retail stores to assist in marketing, advertising, inventory control, predicting faults in telecommunication network. A variety of data mining problems have been studied to help people get an insight into the huge amount of data. One of them is association rule mining, which was first introduced by Agrawal et al. [2]. Agrawal and Srikant [4] extend and define the problem as follows: An itemset is a set of products (items) and a transaction keeps a set of items bought at the same time. The support of an itemset I (denoted as SupI) in a transaction database is the percentage of transactions that contain I in the entire database. An itemset is frequent if its support is not lower than a minimum support threshold (denoted as MST). Association rule mining is to discover all the strong rules in the database. Several methods have been proposed [3], [5], [8], [9], [11].Association rules are usually required to satisfy user specified minimum support and a user specified minimum confidence at the same time. Association rule generation contains two steps:

1. Minimum support is applied to find all frequent item sets in a database.
2. These frequent item sets and the minimum confidence constraint are used to form rules.

II. PROBLEM FORMULATION

Definition 2.1: Association rule mining.

The count of itemset I (denoted as C_I) is the number of transactions containing I in D, and the database size (denoted as |D|) is the number of transactions in D. For two itemsets X and Y , where $X \cap Y = \emptyset$; $X \rightarrow Y$ holds in D (strong rule) if both the following conditions hold, where X and Y are called the precedent and the consequent, respectively.

Manuscript received Dec, 2014.

***Sonali Sambhaji Shintre**, Master of Computer Science & Engineering, T.P.C.T.'s College of Engineering Osmanabad, India.*
***Prof. Pravin P. Kalyankar**, Master of Computer Science & Engineering, T.P.C.T.'s College of Engineering Osmanabad, India.*

1. $Sup_{XUY} = C_{XUY} / |D| \geq MST$ and
2. $Conf_{X \rightarrow Y} = C_{XUY} / C_X \geq MCT$.

Definition 2.2: Association rule hiding
Let D' be the database after applying a sequence of modifications to D. A strong rule X→Y in D will be hidden in D' if one of the following conditions holds in D'

1. $Sup_{XUY} < MST$ and
2. $Conf_{X \rightarrow Y} < MCT$

III. MINING AND HIDING OF SENSITIVE ASSOCIATION RULE

Mining sensitive association rule is one of the recent data mining technique, has already been applied in a wide range of areas. Association rule mining is one of the important problems in the data mining domain. Firstly we have to find out frequent items from large database. Then we have to generate association rules. After that we have to mine sensitive association rules Researchers have recently made efforts at hiding sensitive association rules. It aims at finding interesting patterns among the databases. In association rule mining many useful association rules will be discovered. We have to limit the mining process, in order to keep these rules being hidden. Association rule mining is to find out association rules that satisfy the minimum support and minimum confidence from a given database. This is done by modifying the transactions or items in the database.
Association rule mining is a two-step process.

1. Find all frequent itemsets. By definition, each of these itemsets will occur at least as frequently as a predetermined minimum support.
2. Generate strong association rules from the frequent itemsets. By definition, these rules must satisfy minimum support and minimum confidence.

Association rule mining is decomposed into two sub problems. One is to find those itemsets whose occurrences exceed a predefined threshold in the database, those itemsets are called frequent or large itemsets. The second problem is to generate association rules from those large itemsets with the constraints of minimal confidence.

The classical Apriori algorithm for mining association rule is given below.

```
1.C1= {candidate I-itemsets};
2.L1= {c∊ C1|c.count≥ minsup};
3.FOR (k=2;Lk 1=∅; k++)DO BEGIN
4.Ck=apriori-gen(Lk 1);
5.FOR all transaction t∊ D DO BEDIN
6.Ct=subset(Ck,t);
7.FOR all candidates c∊ Ct DO
8.c.count++;
9.END
10.Lk={c ∊ Ck|c.count≥ minsup}
11.END
12.Answer= L
```

Steps to be followed for hiding sensitive rules:
Step 1: Hide only rules that are supported by disjoint large item sets.
Step 2: Hide association rules by decreasing either their support or their confidence.
Step 3: Select to decrease either the support or the confidence based on the side effects on the information that is not sensitive.
Step 4: One rule to be hiding at a time.
Step 5: Decrease either the support or the confidence, one unit at a time. If an item in XUY is deleted from a transaction containing XUY, SupXUY and Conf XUY will be decreased.

The hiding method includes reducing the support of frequent itemsets containing sensitive rules, reduce the confidence or support of rules. Decreasing confidence of rule involves increasing the support of X in transaction not supporting Y and decreasing the support of Y in transactions supporting both X and Y. Decreasing support of rule involves decreasing the support of the corresponding large itemset. For association rule hiding, two basic approaches have been propose from this we performed on the first approach hides one rule at a time. First selects transactions that contain the items in a given rule. Then it modify the transaction for minimum support and minimum confidence. The modification is done by either removing items from the transaction or inserting new items to the transactions. Then update the original database and association rules are generated, after generating association rules we have to hide sensitive rules. The hiding strategies depends on finding transactions that fully or partially support generating item sets of a rule. Because if a rule has to be hidden, need to decrease the support and confidence of the rule, hiding process is limited.

In order to hide an association rule, X→Y, either decrease its support or its confidence to be less than user specified minimum support threshold (MST) and minimum confidence threshold (MCT). To decrease the confidence of a rule, either increase the support of X, the LHS of the rule, but not support of XUY, or decrease the support of the item set XUY. For the second case, decrease the support of Y, the right hand side of the rule, it would reduce the confidence faster than reducing the support of XUY. To decrease support of an item, the system will modify one item at a time by changing from 1 to 0 or from 0 to 1 in a selected transaction.

The problem of sensitive rule hiding is described as follows:
Given a transaction database, MST, MCT, a set of strong rules, and a set of sensitive items, how we can modify the databases such that using the same MST and MCT, the set of strong rules in the modified database satisfies all the constraints:

1. No sensitive rule.
2. No lost rule.
3. No false rule

Modification Schemes For Rule Hiding
In the following, five modification schemes for rule hiding are introduced, respectively.
Scheme 1: Modify entries from 1s to 0s. As mentioned in [6], if an item in XUY is deleted from a transaction containing XUY, SupXUY and ConfX→Y will be decreased. X→Y is hidden if we repeat this operation until one of the conditions in Definition 2.2 holds
Scheme 2: Modify entries from 0s to 1s. As mentioned in [10], ConfX→Y will be decreased if we insert an item i є X into a transaction that contains X but {i} and does not contain Y. X→ Y is hidden if we repeat this operation until condition 2 in Definition 2.2 holds.

Scheme 3: Modify entries from 1s to 0s or from 0s to 1s.
Scheme 1 can guarantee to satisfy the constraint F -T -H, but Scheme 2 cannot. Both schemes may violate the other two constraints. Scheme 3 alternately uses them to decrease the supports and confidences of sensitive rules. Similarly, this scheme guarantees to satisfy the constraint F -T -H but may violate the others.

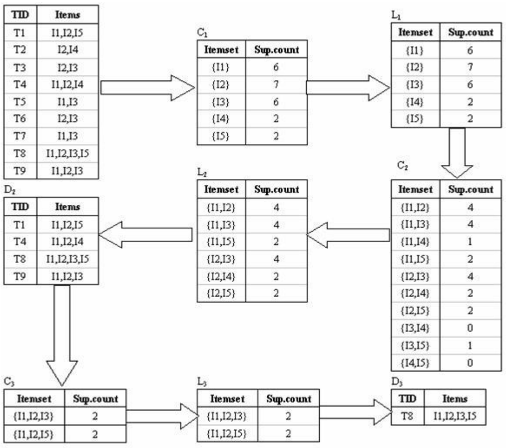
Scheme 4: Change 0s and 1s to ?s. As proposed in [12], for a transaction containing XUY, if an item in XUY is replaced with an unknown, the minimum support of XUY and the minimum confidence of X→Y will be decreased. In addition, for a transaction that contains X but {i} and does not contain Y, if the bit 0 denoting item i is replaced with an unknown, the minimum confidence of X→Y will be decreased. This scheme can guarantee to satisfy the constraint F -T -H but may violate the others.

Scheme 5: Swap 0 and 1 between two transactions. This is a special case of Scheme 3. For each item, the number of entries modified from 1 to 0 must be equal to the number of entries modified from 0 to 1. In this way, the support of each item is unchanged after the rule hiding process. This characteristic can be useful for some applications such as the stock replenishment. However, due to its restriction, this scheme cannot satisfy the three constraints in most cases.

IV. IMPROVED APRIORI ALGORITHM

Apriori is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions. The Apriori Algorithm is the most well-known association rule algorithm and it is used in most commercial products. It uses largest itemset property [14]. “Any subset of a large item set must be large”. The basic idea of Apriori algorithm is to generate item sets of a particular size and then scan the database to count these to see if they are large. Only those candidates that are large are used to generate candidates for the next scan. Association rules are main technique to determine the frequent item set in data mining. The improvement is mainly way of reducing query frequencies and storage resources. We design an improved Apriori algorithm that mines frequent item sets without new candidate generation. For example, in this algorithm we compute the frequency of frequent k- item sets when k-item sets are generated from (k-1)-item sets. If k is greater than the size of transaction T, there is no need to scan transaction T which is generated by (k-1)-item sets according to the nature of Apriori algorithm, and we can remove it.
To implement the improvement in classical Apriori algorithm, the improved algorithm is described as follow steps:

- The function apriori-gen (Lk-1) is called to generate candidate k-item sets by frequent (k-1)-item sets.
- Judging whether C is joined into candidate k-item sets. It is processed by calling function has_infrequent_subset(C, Lk-1). If the return value is true, it means the sets aren’t frequent item sets and should be remove in order to raise efficiency. Otherwise, scan database D.
- The frequency of frequent k-item sets is computed when k-item sets are generated by (k-1)-item sets. If k greater than the size of transaction T, there is no need to scan transaction T which is generated by (k-1)-itemsets and we can use minimum support (MST) for pruning itemsets, and we can delete it. If the size of transaction T is greater than or equal to k, then function subset (Ck,t) is called, which selects only those transaction which are subset of frequent item set. Thus it reduces extra scanning of infrequent itemsets. The detail working of improved Apriori Algorithm is given below in figure 1.



TID	Items
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

C_1

Itemset	Sup.count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

L_1

Itemset	Sup.count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

C_2

Itemset	Sup.count
{I1,I2}	4
{I1,I3}	4
{I1,I4}	1
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2
{I3,I4}	0
{I3,I5}	1
{I4,I5}	0

L_2

Itemset	Sup.count
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2

D_2

TID	Items
T1	I1,I2,I5
T4	I1,I2,I4
T8	I1,I2,I3,I5
T9	I1,I2,I3

C_3

Itemset	Sup.count
{I1,I2,I3}	2
{I1,I2,I5}	2

L_3

Itemset	Sup.count
{I1,I2,I3}	2
{I1,I2,I5}	2

D_3

TID	Items
T8	I1,I2,I3,I5

Fig: 1. Working of Improved Apriori algorithm [13]

We have tested classical Apriori and improved Apriori algorithm on different data set and come with the conclusion that amount of time taken by Improved Apriori algorithm to find frequent itemset is less than classical Apriori algorithm.

V. SYSTEM FRAMEWORK

Following fig.2 shows the framework of our approach that consists of six components. Initially, the original database is converted into the transaction table. The sensitive rule table and the nonsensitive rule table are built to record the rule information. The transaction-rule index is also constructed using the concept of inverted lists [7] to correlate the tables for efficient retrieval.

The main challenge of rule hiding is how to select items and transactions to modify. We propose to represent each class of modifications as a template and then select templates in a one-by-one fashion. The templates are kept in the template table and the selected templates are put into the action table. The six components are updated each time a template is selected. When all the sensitive rules are hidden or the template table is empty, the templates in the action table are applied to modify the original database. If some sensitive rules are not hidden, the user can release as it is, release nothing, or relax the constraint to hide more sensitive rules.

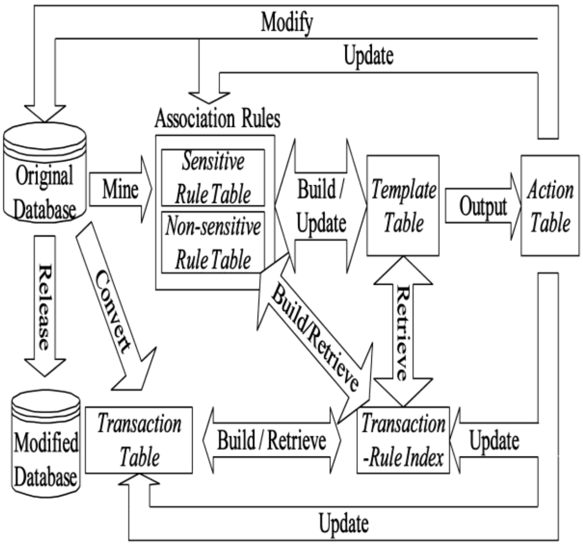
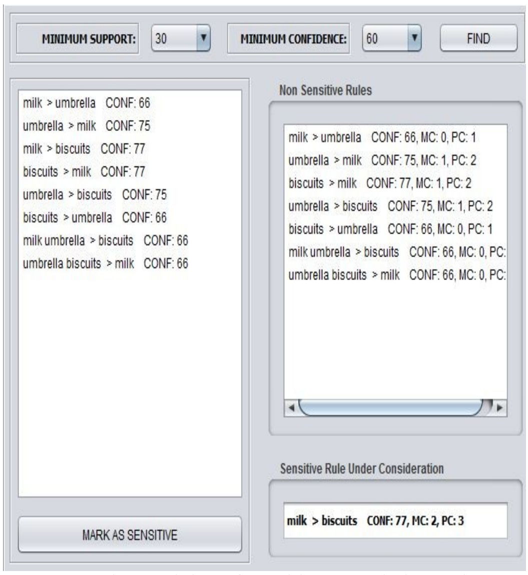


Fig: 2 Framework of our approach

VI. EXPERIMENTAL RESULT

The purpose of this project is the mining and hiding of sensitive association rules by using an Improved Apriori algorithm. Apriori is a classic algorithm for learning association rules. The prairie is designed to operate on the database containing transactions in a transaction database. In this paper frequent item sets are generated from database. Then various association rules will be created under various minimum supports and minimum confidences. We have to mine sensitive association rule after mining these rule select sensitive rule and remaining rules will be nonsensitive In figure 3 Mining of association Rule is shown. Association rule mining scans the database of transactions and calculate the support and confidence of the rules and retrieve only those rules having the support and confidence higher than the user specified minimum support and confidence threshold. In this we have taken Minimum Support 30% and Minimum Confidence 60% and rules are generated shown in Figure 3 having



MINIMUM SUPPORT: 30
MINIMUM CONFIDENCE: 60

```
milk > umbrella   CONF: 66
umbrella > milk   CONF: 75
milk > biscuits   CONF: 77
biscuits > milk   CONF: 77
umbrella > biscuits   CONF: 75
biscuits > umbrella   CONF: 66
milk umbrella > biscuits   CONF: 66
umbrella biscuits > milk   CONF: 66
```

Non Sensitive Rules

```
milk > umbrella   CONF: 66, MC: 0, PC: 1
umbrella > milk   CONF: 75, MC: 1, PC: 2
biscuits > milk   CONF: 77, MC: 1, PC: 2
umbrella > biscuits   CONF: 75, MC: 1, PC: 2
biscuits > umbrella   CONF: 66, MC: 0, PC: 1
milk umbrella > biscuits   CONF: 66, MC: 0, PC:
umbrella biscuits > milk   CONF: 66, MC: 0, PC:
```

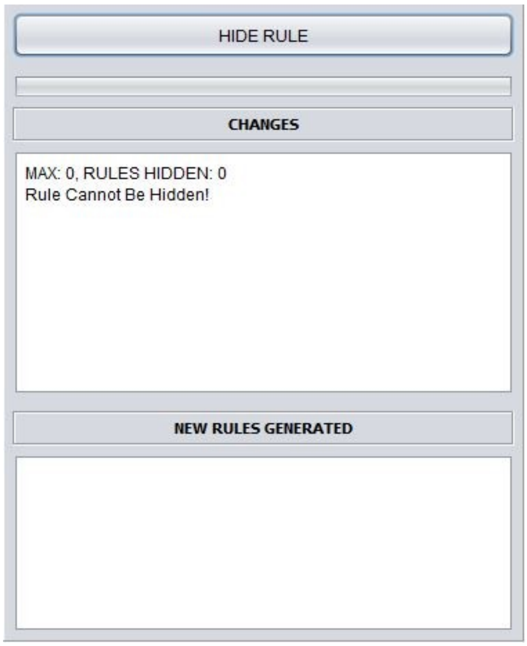
Sensitive Rule Under Consideration

```
milk > biscuits   CONF: 77, MC: 2, PC: 3
```

Fig. 3 Mining of Sensitive Association Rule

Confidence greater than or equal to Minimum Confidence Threshold (MCT). Mark the rule milk => biscuits as sensitive and remaining rules will be nonsensitive rules.

Following Figure 4 shows hiding of sensitive association rule with No Lost Rule and No False Rule. This method shows 0% Lost Rules and no new rules are generated. Then update the original database.



CHANGES

```
MAX: 0, RULES HIDDEN: 0
Rule Cannot Be Hidden!
```

NEW RULES GENERATED

Fig. 4 Hiding of Sensitive Association Rule with No Lost Rule and No False Rule

VII.CONCLUSION

In this paper, we present improved Apriori algorithm to update classical Apriori algorithm. Improved Apriori algorithm generates association rule. The present algorithms can reduces the unnecessary database scan at the time of forming frequent large itemsets and redundancy in the database. Improved apriori algorithm takes less time for generating frequent itemsets as compared to classical apriori algorithm. In this paper we have used novel approach that modifies few transactions in transaction database to decrease support and confidence of sensitive rule without producing side effects. Also the strong rules which are generated by applying improved Apriori algorithm. In this paper we have to generate association rules from large database and then we have to mine sensitive association rules. Above fig. 3 shows the Mining of Sensitive association Rules or marking of sensitive association rule. After mining we have to hide sensitive association rule. Only one rule can be hide at a time. At the time of sensitive rule hiding no false rule, no lost rule should be generated.

ACKNOWLEDGMENT

I would like to express my deep gratitude to my research, Professor Pravin Kalyankar has a research guide, for their patient guidance, enthusiastic encouragement and useful critiques of this research work. I would also like to extend my thanks to the technicians of the laboratory of the Computer department for their help in offering me the resources in running the program.

Finally, I wish to thank my friends and parents for their support and encouragement throughout my project tenure.

REFERENCES

[1] C.M. Chiang, "A New Approach for Sensitive Rule Hiding by Considering Side Effects," master thesis, Dept. of Computer Science, Nat'l Tsing Hua Univ., Republic of China, 2003.

[2] R. Agrawal, T. Imielinski, and A Swami, "Mining Association Rules Between Sets of Items in Large Databases," *Proc. ACM Conf. Management of Data,* pp. 207-216, 1993

[3] R. Agrawal, H. Mannila, R. Srikant, H. Toivinen, and A.I. Verkamo, "Fast Discovery of Association Rules," *Advances in Knowledge Discovery and Data Mining,* chapter 12, U.M. Fayyad et al., AAAI/MIT press, pp. 307-328, 1996.

[4] R. Agrawal and R. Srikant, "Fast algorithm for Mining association Rules," *Proc. Conf. Very Large Data Bases,* pp. 307-328, 1996.

[5] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining Frequent-Patterns without candidate Generation: A Frequent-Pattern Tree Approach*," Data Mining and Knowlede Discovery,* vol. 8.

[6] M. Atallah et. Al., "Disclosure Limitation of Sensitive Rules," *Proc. IEEE Workshop Knowledge and Data Eng. Exchange,* pp. 45-52, 1999.

[7] Information *Storage and Retrieval systems Theory and Implimentation,* G.J. Kowalski and M.T. Maybury, eds. Kluwer Academic Publishers, 1997.

[8] J. Liu, Y. Pan, K. Wang, and J. Han "Mining Frequent Item Sets by Opportunistic Projection," *Proc. ACM Conf. Knowledge Discovery and Data Mining,* 2002.

[9] S. J. Yen and A.L.P. Chen, " A Graph-Based approach for Discovering Various Types of Association Rules," *IEEE Trans. Knowledge and Data Eng.,* vol. 13, no. 5, 2001.

[10] E. Dasseni, V.S. Verykios, A.K. Elmagarmid, and E. Bertino, "Hiding association Rules By Using Confidence and Support," *Proc. Information Hiding Workshop,* pp. 369-383, 2001.

[11] M.J. Zaki, "Scalable Algorithms for Association Mining," *IEEE Trans. Knowledge and Data Eng.,* vol. 12 no. 3, pp. 372-390, 2000.

[12] Y. Saygin, V.S. Verykios, and C. Clifton, "Using Unknowns to Prevent Discovery of association Rules," ACM SIGMOD Record, vol. 30, no. 4, pp. 45-54, 2001.

[13] Sheng Chai et.al, "The Research Of Improved Apriori Algorithm For Mining association Rule", in IEEE 2007b conference.

[14] Lee-Wen Huang, Ye-In Chang, "A Graph Based Approach for Mining Closed large Itemsets" National Sun Yat-Sen University.

[15] Yi-Hung Wu, Chia-Ming Chiang, and Arbee L.P. Chen "Hiding Sensitive Association Rules with Limited Side Effects" IEEE Transaction on Knowledge and data engineering, vol.19, pp. 29-42, 2007.

[16] Shyue-Liang, Bhavesh Parikh, Ayat Jafari "Hiding informative association rule sets" Science direct. 2006.

AUTHOR PROFILE

Sonali Sambhaji Shintre, is a M.E student of Computer Science & Engineering from T.P.C.T.'s College of Engineering, Osmanabad India. She graduated in Information Technology from BAMU University, Aurangabad, India. Her current research work focuses on Efficient Mining & Hiding of Sensitive Association Rules.

Associate Professor Pravin P. Kalyankar, had completed his Master of Computer Science & Engineering, India with Graduate degree in Computer Engineering. Since more than a decade he has been the faculty of Computer science & technology in T.P.C.T.'s College of Engineering, Osmanabad, India where he is currently working has a Head of Department for MCA Department.