International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 13, Number 5 (2017), pp. 833-840
© Research India Publications
http://www.ripublication.com

Sensitive Rule Hiding and InFrequent Filtration through Binary Search Method

Dr. K. Kavitha
Assistant Professor, Department of Computer Science,
Mother Teresa Women's University, Kodaikanal
Tamil Nadu, India.

Abstract

Hiding of sensitive itemsets from extensive information source as an important problem in information exploration. Various methods have been defined which are quite effective but produces only a small variety of frequent items. And also those algorithms cannot control the space complexity. In order to solve and improve the efficiency these of kind of difficulties, the paper proposed an algorithm which integrates improved pruning step and binary search method for hiding sensitive data with minimum number of database scanning.

Keyword: Association Rule, Apriori Algorithm, Minimum Association Rule Mining Support, Minimum Candidate Threshold.

I. INTRODUCTION

Data mining helps to extract interesting patterns from large database. It is an important tool to transform this data into information. Main focus of data mining is to reduce unnecessary database scanning time finding frequent itemset. Mining association rule is one of the main contents of data mining research itemsets and finding the relation of different items in the database. How to generate frequent item sets is a core part in data mining.

Association rule mining selects a database and one of an important issue needs to be considered in finding frequent itemset is to reduce the number of candidate itemsets. In classical Apriori algorithm each time candidate is generated, the algorithm needs to test their occurrence frequency in the database based on their support count.

Mines the database to find rules which satisfys the two threshold support and confidence. The rules are mined using classical Apriori and improved Apriori algorithm and amount of execution time to generate an association rule.

II. RELATED WORK

Novel Pruning Approach for ARM:

Filtration step is introduced in this algorithm which is an alternative step for pruning in apriori perquisite for modified Apriori algorithm is same like classical with the following lemma[1].

Lemma:

If any itemset Z is infrequent then of it superset can be frequent

Proof:

Let Y be an itemset containing all items of Z and number of items in set Y which is greater than Z. (ie) Z < Y. it can be stated as

$$\text{Support}(Z) < \text{minsup} \rightarrow 1$$

It is clear that support(Y) is not greater than any subset became cardinality(Y) is more than Z. (i.e) |Z|<|Y|. It can be written as,

$$\text{Support}(Y) <= \text{Support}(Z) \rightarrow 2$$

Using (1) and (2) of association can be treated as support(Y) < minsup. Otherwise if support of Z is infrequent then Y cannot be frequent.

Procedure:

1. Joining

   Generates candidate set is

   $$C_K = C_K \cup Z$$

2. Filtration

   Removes weak candidate set

   $$P_K = P_K - Y, IF_K = IF_K \cup Y$$

3. Verification

   Generates frequent and infrequent itemset

   $$F_K = F_K \cup Z$$
   $$IF_K = IF_K \cup Z$$

Frequent Itemset Generation Using Binary Search:

Author suggested binary search method, to generate frequent itemsets. Mid value is calculated by low(minimum itemset number) and high(maximum itemset number) value. Based on the mid value, candidate sets are generated.[16]

Approach:

- Input dataset and threshold value
- Calculate length of large itemset in the corresponding dataset
- Apply binary search to find frequent itemsets

```
low = length of small itemset
high = length of large itemset
mid =       (low + high) /2
while(low<=high)
if mid>threshold + length
    low = mid + 1
    high = mid - 1
    Based on the mid value, candidate sets are generated.
```

Mining And Hiding Of Sensitive Association Rule Using Improved Aprori Algorithm:

Improved apriori algorithm to reduce query frequencies and storage resources, without generating new candidate set, frequent itemset are mined using an improved apriori. This algorithm introduces a new method to avoid redundant generation of sub itemset during pruning.[2]

Method:

- Compare minimum support count by minsup * 101
- Produce L_1, candidate set and simultaneously constust TID set for each item during scanning
- Produce L_1, candidate set from joining $L_1 * L_1$
- Delete the patterns whose frequencies are not satisfy minimum support based on threshold limit
- In order to produce L_K join items which satisfy the rule
- Compare with minimum confidence for selecting the hidden rule.

III. PROPOSED METHODOLOGY

This approach integrates the concept Filtration[1] which is alternative to the pruning step in Apriori, Binary Search[16] for reducing the dataset scanning to find frequent itemsets. Hiding which generates sensitive itemset based on minimum confidence.

Lemma:

If count of any itemset Z is less than threshold value then none of it subset can generate candidate set.

Proof:

Low <= high

Let Z be an itemset containing all low length of shortest item and high be the length of longest item for set Z. Can be stated that if candidate set

Low <= high

Mid = (low + high) / 2

It is obvious that count of low cannot be greater than high of any of its subset in a database.

It consists of 5 steps named as

1. Joining – Generate candidate set
2. Pruning – Identify infrequent itemsets
3. Binary Search – Identify the count for generating candidate set
4. Verification – Extract frequent itemsets based on minimum support
5. Hiding – Extract sensitive rules based on minimum confidence

Algorithm:

```
    Modified Apriori(D, minsup, minconf, $F_1$)
Step 1: Binary Search
Low = min(SOT) $\rightarrow$ SOT = Size of Transaction in length of shortest item in D
High = max(SOT) $\rightarrow$ length of longest item in D
    Mid = (low + high) / 2
    While(low <= high)
        K = mid
Step 2: Joining
    $C_k$ is the collection of K-frequent itemsets
    While($F_{k-1} \neq 0$)
    Do $C_k = \emptyset$;
    For(each pair of itemset)
    $\{X_1, X_2, \ldots\ldots\ldots\ldots\ldots\ldots, X_{k-2}, X_{k-1}\} \in F_{k-1}$ and $\{X_1, X_2, \ldots\ldots\ldots\ldots\ldots\ldots, X_{k-2}, Y_{k-1}\}$
    Do if($X_{k-1} < Y_{k-1}$)
```

```
        Then $Z = \{ X_1, X_2, \ldots\ldots\ldots\ldots\ldots\ldots, X_{k-2}, X_{k-1}, Y_{k-1}\}$;
        // Z is the new itemset of size K.
Step 3: Filtration
        $P_k$ is the potential itemset with size K
        $IF_k$ is the collection of K-infrequent itemset
                $P_k = C_k$;
                $IF_k = \emptyset$;
        For(each infrequent itemset $Z \in \cup_{i=2}^{k-1} IF_i$)
                Do if($Z \subset \{Y \mid \exists\, Y \in P_k\}$)
                Then $P_k = P_k - Y$;
                $IF_k = IF_k \cup Y$;
Step 4: Verification
// $F_k$ is the K-frequent itemset collection
        $F_k = \emptyset$
        For(each potential itemset $Z \in P_k$)
        Do if(support(Z) $\geq$ minsup)
        Then $F_k = F_k \cup Z$;
        Else $IF_k = IF_k \cup Z$;
Step 5: Hiding Sensitive Rule
// $S_k$ is the collection of K-sensitive itemset
        $S_k = \emptyset$;
        For(each frequent itemset $Z \in F_k$)
        Do if(confidence(Z) $\geq$ minconf)
        Then $S_k = S_k \cup Z$;
        Else $NS_k = NS_k \cup Z$;
        K = K + 1;
        Print $\cup_{i=1}^{k-2} S_i$;
```

EXPERIMENTAL ANALYSIS

Improvement in proposed algorithm is only way of reducing query frequencies. In this algorithm, frequency of K-itemsets are generated from K-1 items there is no need to scan entire transaction again. Here, infrequent itemsets are eliminated during each candidate set generation. So it goes down the number of database scanning, average time taken(in seconds) by both algorithm as shown below in respect to minimum support value.

Number of transactions = 10,000
Number of items = 100
Average size of transaction is 10
(i.e) |D| = 100K
|T| = 10

Minimum support ranges from 10% to 50% with a step increment of 10%. Results of the experiment are plotted in below figure.

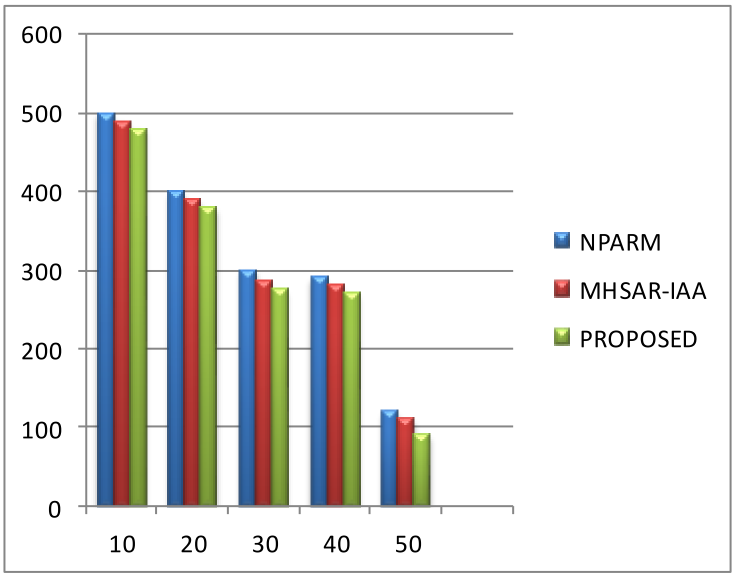

Figure 1: Performance Analysis

All experimental results shows that the proposed algorithm improves better than existing approach.

Performance Analysis of the Proposed Methodology:

1. Main aim of the proposed methodology is to identify sensitive itemset with minimum number of database scanning.
2. Sensitive itemset consists of one or more item.
3. The efficiency of the algorithm is to hide sensitive itemsets which can be measured by minimum confidence through filtration process.

4. When sensitive itemset has more than one item most promising items are considered based on decision making stratergy.
5. Filtration concept helps to identify and separate frequent and infrequent itemsets in order to reduce number of database scanning.
6. In the process of minimizing the database scanning towards Sensitive Data Hiding, Binary Search approach is implemented for satisfying that kind of needs.
7. Binary Search approach helps the administrator to generate relevant candidate set generation as well as avoiding unnecessary itemset generation.

IV. CONCLUSION

In IT technological innovation, gathered information quantity also raised. Data exploration discovers and evaluates the information source to draw out the interesting itemset. This paper has focused a sensitive itemset generation with less number of scanning based on apriori methods and also finds various restrictions of related existing methods. In future, this approach may be implemented in real time application.

REFERENCES

[1] Shintre Sonali Sambhaji, Kalyanakar Pravin P. “Mining and Hiding of Sensitive Association Rule by using Improved Apriori Algorithm”, Proceedings of 18th International Conference, Jan 2015 ISBN: 978-93-84209-82-7.

[2] Lalit Mohan Goyal, M.M.Sufyan Beg and Tanvir Ahmad, “A Novel Approach for Association Rule Mining”, International Journal of Information Technology, Vol 7, Issn 0973-5658, Jan 2015.

[3] Agrawal, R., Imielinski, T., and Swami, A. N. 1993. Mining Assocation rules between sets of Items in large databases. In Proceedings of the 1993 aCM SIGMOD International Conference on Management of Data, 207-216.

[4] Agrawal, R., Srikant R, Fast algorithms for mining association rules, Proceedings of the 20th International Conference on Very large databases, New York: IEEE press, PP. 487-499,1994

[5] S. Brin, etc – Dynamic itemset counting and implication rules for market based analysis, Proceedings ACM SIGMOD International Conference of Data, PP. 255-264, 1994.

[6] M. J. Zaki, “Scalable algorithms for assoiction mining”, IEEE transaction on knowledge and data engineering, pp.372-390, 2000.

[7] J. Han, J. Pei, Y. Yin- Mining frequent patterns without candidate generation. Proceedings of SIGMOD 2000.

[8] F.C.Tseng and C.C. Hsu – Generating frequent patterns with the frequent pattern list, Proc. 5th Pacific Asia Conf. on Knowledge Discovery and Data Mining, pp.376-386, April 2001.

[9] Nicolas pasquier, yves bastide, rafik taouil, and lotfi lakhal. Discovering frequent closed itemsets for association rules. In proc. ICDT 99, pages 398-416, 1999.

[10] M.J.Zaki and C.Hsiano, “Charm: An efficient algorithm for closed itemset mining”, Proc SIAM international conference Data Mining, pp. 457-473, 2002.

[11] J. Pei, J. Han and R. Mao, “Closet: An efficient algorithm for mining frequent closed itemsets”, ACM SIGMOD workshop research issue in Data mining and knowledge discovery, pp. 21-30, 2000.

[12] J.Wang, J.Han and J.Pei “Closet: Searching for the best strategies for mining frequent closed itemsets”, proc. Intl conf. knowledge discovery and data mining, pp. 236-245, 2003.

[13] G. Grahne, and J.Zhu “Effciently using prefix trees in mining frequent itemsets”, IEEE ICDM Workshop on Frequent Itemset Mining Implementations, 2003.

[14] C.Lucchese, S.Orlando and R. Perego, “DCI Closed: a fast and memory efficient algorithm to mine frequent closed itemsets”, IEEE Transaction on Knowledge and Data Engineering, Vol 18, No.1 PP.21-35, 2006

[15] D.Lin, Z.M.Kedem, Pincer-Search: A new algorithm for discovering the maximum frequent itemset, EDBT Conference Proceedings, 1998, pages 105-110.

[16] Manisha Kundal and Dr. Parminder Kaur, “Various Frequent Item Set on Data Mining Technique”, Global Journal of Computer Science and Technology Software and Data Engineering, Vol 15 issue 4 version 1.0, ISSN: 0975-4172, 2015