ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2015

A Time & Memory Efficient Technique for Mining Frequent Pattern Mining

Pradeep Rupayla[1], Kamlesh Patidar[2]

PG Scholar, Dept of Computer Science, JIT Borawan, Khargone, MadhyaPradesh, India[1]

Assistant Professor, Dept of Computer Science, JIT Borawan, Khargone, MadhyaPradesh, India[2]

ABSTRACT: Frequent item set mining is one of the most popular field and most common field of data mining. At the same time, it is a very complex and a time consuming process. Although there are many algorithms are available to mine the frequent patterns from a voluminous data set, but there is still a lot of scope to mine frequent data from different data sets in less time & in less memory. Frequent pattern mining is very useful in cross marketing, market basket analysis, credit card fraud detection. Knowledge discovery in databases (KDD) helps to identifying precious information in such huge databases. This information helps the decision makers in making decision. Ultimately this type of information helps in various goals like – sales increase, profit maximization, prediction etc. In this paper, we have proposed a novel compact data structure based method to discover frequent pattern mining. The proposed method transforms the original data set into a transformed and compacted data set & then it discovers the frequent patterns from the transformed data set.

KEYWORDS: Data Mining, Association Rule, Support, Confidence, Frequent Item-sets.

I. INTRODUCTION

Data mining is vital in many areas like market basket analysis, web usage mining, credit card fraud detection etc. The newly extracted information or knowledge may be applied to information management, query processing, process control, decision making.

Data mining represents the integration of several fields. Data mining can be defined as a non-trivial process of identifying.

- Valid
- Novel
- potentially useful
- ultimately understandable Patterns in data. It employs techniques from
- machine learning
- statistics
- databases

Association rule is an implication of the form X -> Y where X,Y subset of I are the sets of items called Item sets and X ∩Y = Φ. Association rules show attributes value conditions that occur frequently together in a given dataset. A commonly used example of association rule mining is Market Basket Analysis [2]. We use a small example from the supermarket domain. The set of items is-

I = {Milk, Bread, Butter, Beer}

A rule for the shopping market could be **{Butter, Bread} =>{Milk}**meaning that if butter and bread are bought, customers also buy milk. They could use this data for adjusting store layouts (placing items optimally with respect to each other), for cross- selling, for promotions, for catalogue design and to identify customer segments based on buying patterns.

Association rules provide information in the form of "if-then" statements. These rules are computed from the data and, unlike the if-then rules of logic, association rules are probabilistic in nature. If 90% of transactions that purchase bread and butter, then also purchase milk.

Antecedent: bread and butter

10.15680/ijircce.2015.0302081

Consequent: milk
Confidence factor: 90%

In addition to the antecedent (the "if" part) and the consequent (the "then" part), an association rule has two numbers that express the degree of uncertainty about the rule.In association analysis the antecedent and consequent are sets of items (called item sets) that are disjoint (do not have any items in common).

Support for an association rule X->Y is the percentage of transaction in database that contains X U Y . Confidence or Strength for an association rule X U Y is the ratio of number of transactions that contains X U Y to number of transaction that contains X.An itemset (or a pattern) is frequent if its support is equal to or more than a user specified minimum support (a statement of generality of the discovered association rules).Association rule mining is to identify all rules meeting user specified constraints such as minimum support and minimum confidence (a statement of predictive ability of the discovered rules). One key step of association mining is frequent itemset (pattern) mining, which is to mine all itemsets satisfying user specified minimum support. [10].However a large number of these rules will be pruned after applying the support and confidence thresholds. Therefore the previous computations will be wasted. To avoid this problem and to improve the performance of the rule discovery algorithm, mining association rules may be decomposed into two phases:

1. Discover the large itemsets, i.e., the sets of items that have transaction support above a predetermined minimum threshold known as frequent Itemsets.
2. Use the large itemsets to generate the association rules for the database that have confidence above a predetermined minimum threshold

Frequent item set mining plays an important role in several data mining fields as association rules [1,2,4] warehousing [10], correlations, clustering of high-dimensional biological data, and classification [9]. Given a data set d that contains k items, the number of itemsets that could be generated is 2k - 1, excluding the empty set [1]. In order to searching the frequent itemsets, the support of each item set must be computed by scanning each transaction in the dataset. In addition another new algorithm has been developed [5] which uses top down graph based approach. In addition, many researches have been developed algorithms using tree structure, such as H-mine [3], FP-growth [6], and AFP-Tree [7].

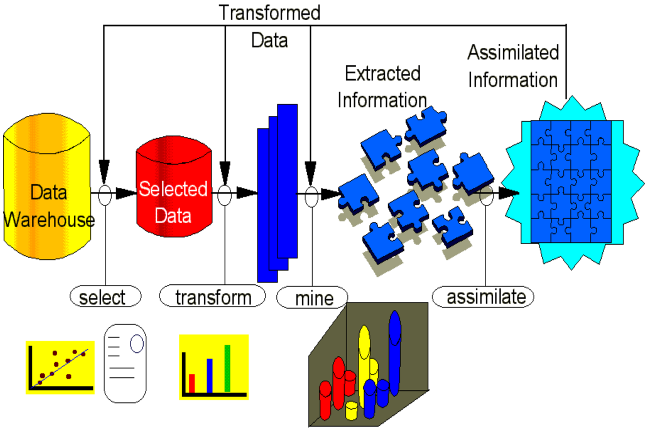


Figure 1.1: key steps in data mining

II. Literature Survey

Aggrawal.R, Imielinski.t, Swami.A[2] defined the problem of finding the association rules from the database. Association rule mining can be defined formally as follows: Association rule is an implication of the form X -> Y where X,Y subset of I are the sets of items called Item sets and X ∩Y = Φ. Association rules show attributes value conditions that occur frequently together in a given dataset. A commonly used example of association rule mining is Market Basket Analysis [2]. One key step of association mining is frequent item set (pattern) mining, which is to mine all itemsets satisfying user specified minimum support [14]. As frequent data itemsets mining are very important in mining the Association rules. Therefore there are various techniques are proposed for generating frequent itemsets so that association rules are mined efficiently. The approaches of generating frequent itemsets are divided into basic three techniques.

1.Horizontal layout based data mining techniques.

- Apriori algorithm.
- DHP algorithm.
- Partition.
- Sample.
- A new improved Apriori algorithm.

2.Vertical layout based data mining techniques.

- Eclat algorithm.

3.Projected database based data mining techniques.

- FP-tree algorithm.
- H-mine algorithm.

There are dozens of algorithms used to mine frequent itemsets. Some of them, very well known, started a whole new era in data mining. They made the concept of mining frequent itemsets and association rules possible.

The most popular frequent itemset mining called the FP-Growth algorithm was introduced by [5]. The main aim of this algorithm was to remove the bottlenecks of the Apriori-Algorithm in generating and testing candidate set.

Eclat [3,11,14] algorithm is basically a depth-first search algorithm using set intersection. It uses a vertical database layout i.e. instead of explicitly listing all transactions; each item is stored together with its cover (also called tid list) and uses the intersection based approach to compute the support of an itemset.

The SaM (Split and Merge) algorithm established by [10] is a simplification of the already fairly simple RElim (Recursive Elimination) algorithm. While RElim represents a (conditional) database by storing one transaction list for each item (partially vertical representation), the split and merge algorithm employs only a single transaction list (purely horizontal representation), stored as an array.

Partitioning algorithm [1] is based to find the frequent elements on the basis partitioning of database in n parts. It overcomes the memory problem for large database which do not fit into main memory because small parts of database easily fit into main memory.

It was absorbed in [12,13] that the improved algorithm is based on the combination of forward scan and reverse scan of a given database. If certain conditions are satisfied, the improved algorithm can greatly reduce the iteration, scanning times required for the discovery of candidate itemsets. Suppose the itemset is frequent, all of its nonempty subsets are frequent, contradictory to the given condition that one nonempty subset is not frequent, the itemset is not frequent. Based on this thought, proposes an improved method by combining forward and reverse thinking: find the maximum frequent itemsets from the maximum itemset firstly, then, get all the nonempty subsets of the frequent itemset.

III. PROPOSED ALGORITHM

Input:

- A Transaction Database TDB
- MST – Minimum support Threshold

Step1: Scan the data base (TDB) to find the support count of each single item. Store this result in a new data structure called Table.

Step2: Compare the support of each element of Table to the minimum threshold. If the support of any element is less then the minimum threshold then that element is discarded. Now arrange all the elements of Table in the decreasing order of their support count.

Step 3: Discard all the infrequent item found in step2 are discarded from the original TDB. In this way, we will get a new NTDB, whose transaction will contain elements with support count greater than the threshold. Now rearrange all the transactions of NTDB in the decreasing order of their item count.

Step 4: Store all the transactions and their count in a multidimensional table (MTable). Then select transaction of highest size whose count is greater than the minimum threshold. If no such transaction found then select highest sized

and second highest sized transaction to generate the second highest sized item set. Continue this process until frequent item sets with greater support count are found.
Step 5: Apply A Priori property on the result of step 4.
Step 6: Scan the NTDB to locate the item which is frequent but still not included in the list of frequent item sets. Reduce the NTDB according to those items. It is called Left Over Transaction Data Base (LOTDB).
Step 7: If no such transaction exists in NTDB then go to step 8 else repeat step 3 to 7
Step 8: Halt.
Output: A Set fo Frequent Elements.

IV. Result Analysis

Input Data Set

The input data set is as follows:

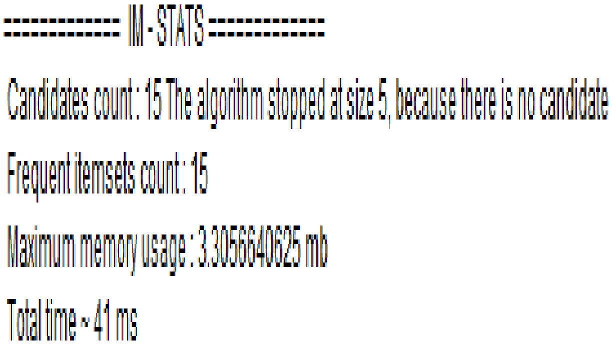
1 3 4
2 3 5
1 2 3 5
2 5
1 2 3 5
1 3 4
2 3 5
1 2 3 5
2 5
1 2 3 5
The MST is 40%.

Previous Algorithm:

The results of the algorithm are as follows:

Output:

```
1 supp: 6

2 supp: 8

3 supp: 8

5 supp: 8

1 2  supp: 4

1 3  supp: 6

1 5  supp: 4

2 3  supp: 6

2 5  supp: 8

3 5  supp: 6

1 2 3  supp: 4

1 2 5  supp: 4

1 3 5  supp: 4

2 3 5  supp: 6

1 2 3 5  supp: 4
```



```
============= IM - STATS =============

Candidates count : 15 The algorithm stopped at size 5, because there is no candidate

Frequent itemsets count : 15

Maximum memory usage : 3.3056640625 mb

Total time ~ 41 ms
```

Figure 3.1:Time and space consumed by previous algorithm.

Proposed Algorithm:

The results of the algorithm are as follows

1 supp :6

2 1 supp:4

3 1 2 supp:4

5 1 2 3 supp:4

5 1 2 supp:4

3 1 supp:6

5 1 3 supp:4

5 1 supp:4

2 supp:8

3 2 supp:6

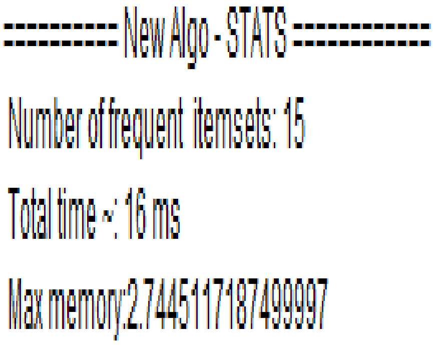
5 2 3 supp:6

5 2 supp:8

3 supp :8

5 3 supp:6

5 supp :8



```
========== New Algo - STATS ===========

Number of frequent  itemsets: 15

Total time ~: 16 ms

Max memory:2.7445117187499997
```

Figure 3.2:Time and space consumed by proposed algorithm.

Graphical Representation of Result Analysis:-

1. Memory comparison of Previous and Proposed Algorithm:

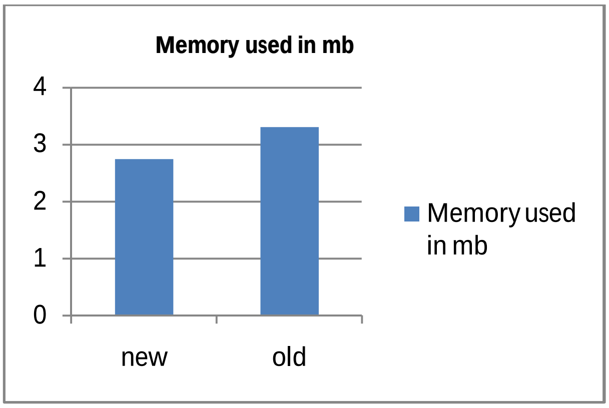


Figure3.3:Memory Comparison

2. Frequent item-set found by Previous and Proposed Algorithm:

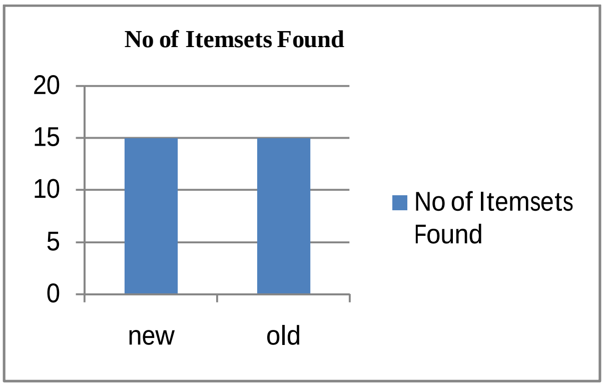


Figure 3.4:Result Comparison(itemset found)

1. Time wise comparison of Previous and Proposed Algorithm:

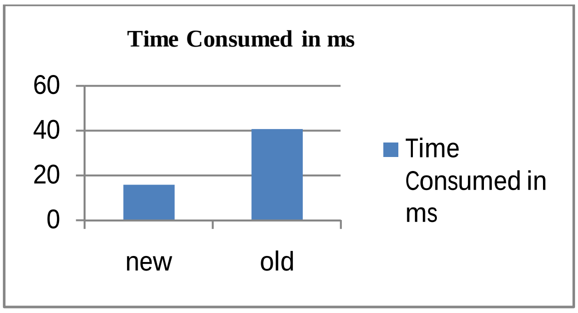


Figure 3.5:Time Comparison

V. CONCLUSION

Data mining is the heart favourite topic for many young scientists. This is because it has been widely used in many real world applications, ranging from the cross marketing to disease prediction. It is also used in the making of the expert systems. This paper proposed an enhanced data mining technique. This technique fetches all the relevant frequent patterns from a large data set by using less CPU time & space in comparison to the existing method. The result comparison also shows that the proposed work finds the frequent patterns in less time.

.

REFERENCES

1. A. Savasere, E. Omiecinski, and S. Navathe. “An efficient algorithm for mining association rules in large databases”. In Proc. Int’l Conf. Very Large Data Bases (VLDB), Sept. 1995, pages 432–443.
2. Aggrawal.R, Imielinski.t, Swami.A. “Mining Association Rules between Sets of Items in Large Databases”. In Proc. Int’l Conf. of the 1993 ACM SIGMOD Conference Washington DC, USA.
3. Agrawal.R and Srikant.R. “Fast algorithms for mining association rules”. In Proc. Int’l Conf. Very Large Data Bases (VLDB), Sept. 1994, pages 487–499.
4. Brin.S, Motwani. R, Ullman. J.D, and S. Tsur. “Dynamic itemset counting and implication rules for market basket analysis”. In Proc. ACM-SIGMOD Int’l Conf. Management of Data (SIGMOD), May 1997, pages 255–264.
5. C. Borgelt. “An Implementation of the FP- growth Algorithm”. Proc. Workshop Open Software for Data Mining, 1–5.ACMPress, New York, NY, USA 2005.
6. Han.J, Pei.J, and Yin. Y. “Mining frequent patterns without candidate generation”. In Proc. ACM-SIGMOD Int’l Conf. Management of Data (SIGMOD), 2000.
7. Park. J. S, M.S. Chen, P.S. Yu. “An effective hash-based algorithm for mining association rules”. In Proc. ACM-SIGMOD Int’l Conf. Management of Data (SIGMOD), San Jose, CA, May 1995, pages 175–186.
8. Pei.J, Han.J, Lu.H, Nishio.S. Tang. S. and Yang. D. “H-mine: Hyper-structure mining of frequent patterns in large databases”. In Proc. Int’l Conf. Data Mining (ICDM), November 2001.
9. C.Borgelt. “Efficient Implementations of Apriori and Eclat”. In Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations, CEUR Workshop Proceedings 90, Aachen, Germany 2003.
10. Toivonen.H. “Sampling large databases for association rules”. In Proc. Int’l Conf. Very Large Data Bases (VLDB), Sept. 1996, Bombay, India, pages 134–145.
11. Nizar R.Mabrouken, C.I.Ezeife. Taxonomy of Sequential Pattern Mining Algorithm”. In Proc. in ACM Computing Surveys, Vol 43, No 1, Article 3, November 2010.
12. Dongme Sun, Shaohua Teng, Wei Zhang, Haibin Zhu, “An Algorithm to Improve the Effectiveness of Apriori”. In Proc. Int’l Conf. on 6th IEEE Int. Conf. on Cognitive Informatics (ICCI'07), 2007.pp-66-70.
13. Jiawei Han, Micheline Kamber, Morgan Kaufmann “Data mining Concepts and Techniques” Publishers, 2006
14. Pei.J, Han.J, Lu.H, Nishio.S. Tang. S. and Yang. D. “H-mine: Hyper-structure mining of frequent patterns in large databases”. In Proc. Int’l Conf. Data Mining (ICDM),2001.