Result Analysis of Mining Fast Frequent Itemset Using Compacted Data

Shweta Kharat and Neetesh Gupta

Information technology Department, TIT, Bhopal (M.P.) India

ABSTRACT

Data mining and knowledge discovery of database is magnetizing wide array of non-trivial research arena, making easy to industrial decision support systems and continues to expand even beyond imagination in one such promising field like Artificial Intelligence and facing the real world challenges. Association rules forms an important paradigm in the field of data mining for various databases like transactional database, time-series database, spatial, object-oriented databases etc. The burgeoning amount of data in multiple heterogeneous sources coalesces with the impediment in building and preserving central vital repositories compels the need for effectual distributive mining techniques.

The majority of the previous studies rely on an Apriori-like candidate set generation-and-test approach. For these applications, these forms of aged techniques are found to be quite expensive, sluggish and highly subjective in case there exists long length patterns.

KEYWORD

Data Mining, Frequent pattern mining, Apriori Algorithm.

1. INTRODUCTION

The mining paradigms and the robustness of many approaches are provided to aid the performance issues arises in various proposed algorithms yet. A large number of algorithms have been suggested for frequent item set mining during the last decade Most of the algorithms share the same general approach: generate a set of candidate item sets, count up their frequencies in D, and use the obtained information in generating supplementary candidates, until the complete set is found. The methods differ primarily in the order and extent of candidate generation. The most famous is probably the Apriori algorithm, developed independently by Agrawal et al. [1]. Frequent pattern mining is the process of searching recurring relationships in a given dataset. It leads to Most of the other algorithms are the variant of Apriori Algorithm which is the foundation of many such related approaches.

Let I = {i1, i2, …, im} be a set of items. Let D, the task relevant data, be a set of database transactions where each transaction T is a set of items such that T€ I.

Let A be a set of items. A transaction T is said to contain A if and only if A € T. An association rule is implication of the form A=>B, where A, B are set of I and A ∩ B = Φ. The rule A=>B holds in the transaction set D with support 's', where s is the percentage of transactions in D that contain A U B (i.e., both A and B). The rule A => B has confidence c in the transaction set D if c is the percentage of transactions in D containing A that also have B. That is,

Support (A & B) = P(A U B) (1) confidence (A & B) = P(B | A) (2)

The goal of the frequent pattern mining is to identify all the set of items that satisfy minimum support.

2. EXISTING WORK

Apriori algorithm

This is the classical algorithm used for mining frequent patterns and was proposed by R. Agrawal [1]. This algorithm runs in several passes. The first pass of the algorithm simply counts item occurrences to determine the large 1-itemsets. A subsequent pass, say pass k, contains of two phases. First, the large itemsets Lk-I found in the (k-1)th pass are used to generate the candidate itemsets Ck, by using the Apriori candidate generation function (apriori-gen). Next, the databank is scanned and the support of candidates in Ck is calculated. The Apriori algorithm is:

```
L1 = {large 1-itemsets};
for ( k = 2; Lk-1 ≠ Ø; k++ ) do begin Ck = apriori-gen(Lk-1);
//New candidates

For all transactions t € D do begin Ct = subset(Ck, t);
//Candidates contained in t for all candidates c € Ct do c.count++;
end Lk = { c € Ck | c.count ≥ minsup }
end

Answer = Ck U Lk;
```

The apriori-gen function takes as argument Lk-1, the set of all large (k-1) itemsets. It incomes a superset of the set of all large k-itemsets and these item sets are treated as candidates and only for these candidates the support is counted.

3. REVERSE APRIORI[15]

This approach is reverse of the Apriori one. Here in reverse Apriori, it generates large frequent item sets which starts by considering a maximum combination of all the values in pairs. It is efficient in the case it is obvious to find out these combinations by having a glance at it and generating a minimum support value in the dataset. It then generates a large frequently mined set of items on the condition that it should satisfy the user defined support. If it does, then it gradually decreases the number of items in the item sets unless it gets the largest set of frequent items. In this way, reverse Apriori algorithm works in an opposite fashion giving the frequently mined item sets. The pseudo code for reverse apriori is as follows:

Input :

Database D

Minimum Support ε

Output:

Large Frequent Itemsets

Method:

```
01   K= maximum number of attributes
02  I=0
03    For all combinations of (k-1) number of   attributes
04  Do
05  generate candidate-K-1 itemsets
    generate frequent itemsets from
06  candidate-
    K-1 itemsets
07  where,  support  count  of  generated
    itemsets >= ε
08  If successful, then go to step 10
9     Else i=i+1 and go to step 03
10   Return sets of large frequent itemsets
11   End
```

Reverse Apriori begins by calculating total number of attributes and is stored in a variable k and an incremental variable is initialized to zero. In further steps, all the combinations are checked by predefined support count. If it satisfies the support count then the combination is a member of frequent itemset and the itemset is the large frequent itemset. If the steps 3-7 fails to generate large frequent itemset then the step 9 is incremented by 1 and will jump back to step 3. This process continues until a large frequent itemset is generated. When the results are obtained, then the program jumps to step 10 where the output exists. Though reverse Apriori works better than Apriori but still there are chances of getting some irrelevant patterns while finding out the maximum large frequent itemset in k.

4. PROPOSED ALGORITHM

Bottom up frequent pattern mining:

There are basically two approaches that work with every procedure; either it is top down or bottom up. Apriori is a top down one. Unlike Apriori which first tries to find candidate-1 itemsets and begins to prune those generated candidate-1 itemsets which doesn't have the minimum support value. Then subsequent candidate itemsets are generated from the one and the same process continues to iterate itself until the maximum frequent item sets are generated from the entire database. This process works efficiently but performance start degrading when the database grows intermittently.

This approach works in a completely opposite manner which is bottom-up and in this way it differs itself from Apriori algorithm. In this paper how it works and how can it be proven more efficient than Apriori. In this approach, first find out the conjunctive pattern by making all possible pairs of itemset and discard the items which does not satisfy the user defines minimum support and evaluate a maximum possible limit of number of items in the dataset thereby generating a huge bulk of frequent item sets satisfying a user specified minimum item support. It will subsequently decrease the conjunctive frequent item set till it acquires a set of possible frequent item sets.

Conjunctive patterns are described as the patterns containing the most simultaneously occurred item sets called as frequent item set.

Let d= (A, B, C, D) are the item set where A, B, C, D belongs to the transaction t. the pairs are said to be conjunctive if (A,B) E user defined support.

A pattern I is said to be frequent if Supp (I) is greater than or equal to a minimum support threshold, denoted min supp.

On the contrary, disjunctive patterns are those which contain all the different and irrelevant pairs of sets and therefore should be discarded as they are outliers

Disjunctive if (A, B)! E user defined support

Let’s consider an example to understand this concept: In much generalized terms, let’s consider a transaction based on a super market which contains a huge set of item sets and their occurrences. It has been zeroed in user defined support on milk-made items. Considering a transaction that has all the possibilities of items being paired, now to make this approach work faster by shedding light to the concept of conjunctive and disjunctive patterns which are defined above. Now this transaction consists of all the items ranging from –

T = {bread, onion, banana, butter, toothpaste, cheese, egg, pasteurized milk, peas, wafers, biscuits ……..}

Now user defined support is already fixed to milk-made products then it’s not a productive step to take sample combination of all the item sets one by one and then generate candidate-1 item sets then frequent-1 item sets and so on. Thus what can be done here is, it just take only those item sets into account which seem to lie in this category of user defined support and that is milk made products.Thus the conjunctive pattern will contain only those products which fall into this specified range. And the rest of the items are considered as disjunctive patterns since they do not fall under the category of selection and therefore needs to be discarded.

Conjunctive sets = {cheese, pasteurized milk, butter}

Disjunctive sets = {bread, egg, toothpaste, wafers}

The reverse Apriori [15] is then applied which works faster than the existing Apriori algorithm and subsequently all the frequent item sets are produced by lessening the amount of candidate generations.

One more example can be taken to understand it more deeply. Here is one example to understand this approach through the subsets of the item sets.-

ITEMSET	**VALUES OF ITEMSET**
Temperature	Hot, mild, chilling
Humidity	High, normal, low
Pitch	Dry, damp
Soccer	Yes, no

Here let's assume that Jack has a fixed user defined support of playing the soccer if and only if the weather is mild and dry. Then by declining the combination of irrelevant and unnecessary items and their values is an effective way to reach onto the decision than by considering all in all sets.

Through these very simple and easy to understand examples, the conjunctive and disjunctive pattern are getting deployed and how they can be diminish the need of higher order candidate generation procedure. The proposed bottom-up algorithm with conjunctive pattern is:

Input:

A database D containing transactions T.

Min_support S

Output:
Large frequent item set

Algorithm:

```
1. Scan the database transaction which has some distinct items

   T = {X, Y, Z, F, P, M, L, S}

2. Find out the conjunctive patterns from the transaction

   If X, Y, F € usr-def-sup

3. Conj = ( X, Y, F ) Else

   Disj = {P, L, M, Z}
4. Max=con

5. j=0
```

```
6. For all further combinations of (max-i) number of attributes

7. Do

8. Generate candidate (max-i) item sets

9. Frequent (max-i) item sets FPk is generated from candidate (max-i) item sets

10. Where support count of generated item sets >= min_sup

11. If successful then go to step13

12. Else j=j+1 and go to step 6

13. Return sets of large frequent item sets
14. End
```

4. EXPERIMENTAL EVALUATION AND PERFORMANCE STUDY

To calculate the efficiency and effectiveness of the improved algorithm, two algorithms has been performed: Apriori algorithm and Modified Apriori algorithm. All the experiments were examined on Pentium IV Machine, running Microsoft windows 7.The algorithm is implemented in Java1.7.
Comparison of the two algorithms on two different measures. Those are mentioned below:

(a) Time requirement
(b) Memory usage

By executing both algorithms on retail database which has 80,000 of records for comparing their results on the basis of time and memory usage. In the following tables, the results are obtained from the experiments. Then comparative graphs are plotted using these statistics to get a comparative result among the two.

Here is the table: Time requirement

Support (%)	**Apriori (msec)**	**Modified Apriori (msec)**
10	483	468
20	468	453
30	484	453
40	546	500
50	530	452

Table 1Time requirement of Apriori and Modified Apriori

In the above table time requirement of both the algorithms are shown at different support values. Next graphs have been plotted for the time requirements of two algorithms against number of transactions in the database. The graphs are shown below:

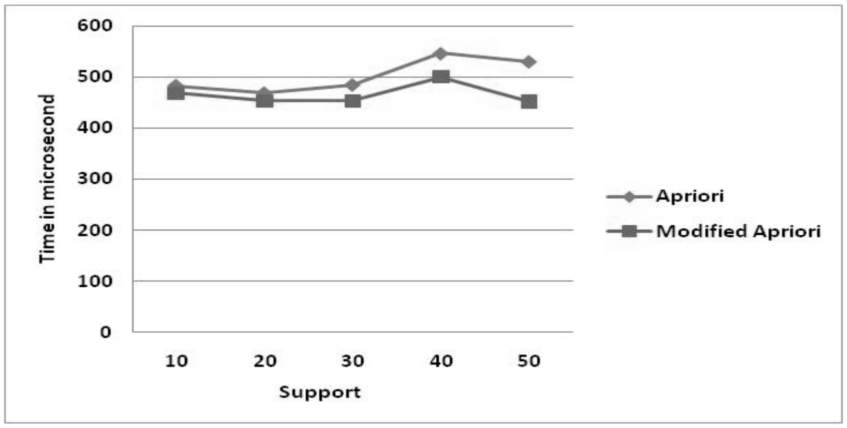

Fig.1 Time requirement of Apriori and Modified Apriori

Here is the table: Memory usage

Support (%)	**Apriori (msec)**	**Modified Apriori (msec)**
10	12.10	11.53
20	12.77	11.97
30	12.23	11.94
40	15.15	12.93
50	13.26	12.71

Table 2 Memory Usage of Apriori and Modified Apriori

In the above table memory usage of both the algorithms are shown at different support values.

Next graphs have been plotted for the memory usage of two algorithms against number of transactions in the database. The graphs are shown below:

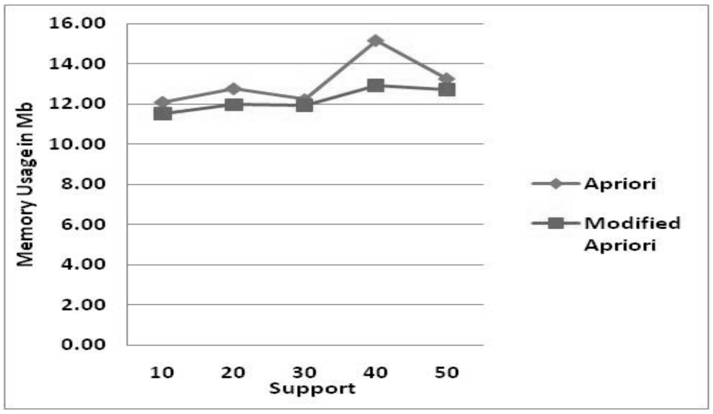

Fig.2 Memory Usage of Apriori and Modified Apriori

From the above tables and the graphs, it has been proved that:

Modified Apriori works far better than Apriori algorithm in respect of time required and memory usage for execution.

5. Conclusion

Discovering association rules and frequent pattern mining have an ample space to propound deep and many efficient algorithms have been proposed up to now, but applying those rules still found to be computationally expensive. An Apriori and reverse Apriori were suggested for frequent item set mining. Reverse Apriori is partly refined in this study. Instead of pairing all the frequent item sets at the end, which results in generating even higher number of generated candidate sets, it is viable to collect a heavy collection of frequent item set and then pruning it out thus giving lesser scans. This proposed method provides useful assistance is generating frequently mined item sets through the use of concept of conjunctive pattern mining that reflects the important information that transforms a raw data into a useful set of frequent itemsets with less candidates generated. Also there are least chances of getting irrelevant item sets as compared to reverse Apriori and removing this bottleneck of reverse Apriori. The main advantage of an improved algorithm is that it can reduce the time for execution of Apriori algorithm.

References

[1] Agrawal, R., Imielinski, T., and Swami, A. N. 1993. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 207-216.
[2] Kamber and Han, “Data Mining Concepts and Techniques”, Hartcourt India P. Ltd., 2001.
[3] Arun K Pujari (2003). Data Mining Techniques (Edition5): Hyderabad, India: Universities Press (India) Private Limited.
[4] Syed Khairuzzaman Tanbeer , Chowdhury , Farhan Ahmed and Byeong-Soo Jeong , Parallel and Distributed Algorithms for FP mining in large Databases, IETE Technical Review Vol 26, Issue 1 , pp 55-65, Jan 2009.

[5] Renata Ivancsy and Istvan Vajk, Fast Discovery of Frequent Itemsets : a cubic structure based Approach , Information 29, pp 71-78, 2005.
[6] R.J. Bayardo, “Efficiently Mining Long Patterns from Databases”, In L.M. Haas and A. Tiwary (eds.), Proc. Of the ACM SIGMOD Int. Conf. on Management of Data, June 1998, pp. 85-93.
[7] B. Goethals, “Efficient Frequent Pattern Mining”, PhD thesis, University of Limburg, Belgium, Dec. 2002.
[8] Z. Zheng, R. Kohavi, and L. Mason, “Real World Performance of Association Rule Algorithms”, In F. Provost and R. Srikant (eds.), Proc. of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 401-406.
[9] Kryszkiewicz, M.: Concise representations of association rules. In: Proceedings of the ESF ExploratoryWorkshop on Pattern Detection and Discovery in Data Mining, Springer-Verlag, LNCS, volume 2447. (2002) 92–110
[10] Nanavati, A.A., Chitrapura, K.P., Joshi, S., Krishnapuram, R.: Mining generalised disjunctive association rules. In: Proceedings of the 10th International Conference on Information and Knowledge Management. (2001) 482–500.
[11] H´ajek, P., Havr´anek, T.: Mechanizing Hypothesis Formation: Mathematical Foundations for a General Theory. Springer-Verlag (1978).
[12] Denden, I., Hamrouni, T., Ben Yahia, S.: Efficient exploration of the disjunctive lattice towards extracting concise representations of frequent patterns. To appear in the Proceedings of the 9th African Conference on Research in Computer Science and Applied Mathematics (in French). (2008).
[13] Valtchev, P., Missaoui, R., Lebrun, P.: A fast algorithm for building the Hasse diagram of a Galois lattice. In: Proceedings of the Conference on Combinatorics, Computer Science and Applications. (2000)
[14] Boulicaut, J.F., Bykowski, A., Rigotti, C.: Free-sets: A condensed representation of Boolean data for the approximation of frequency queries. Data Mining and Knowledge Discovery volume 7(1) (2003) 5–22
[15] Kamrul Abedin Tarafder , Shah Mostafa Khaled , Mohammad Ashraful Islam , Khandakar Rafiqual Islam , Hasnain Feroze , Mohammed Khalaquzzaman and Abu Ahmed Ferdaus , 2008. “Reverse Apriori Algorithm for Frequent Pattern Mining.”

Authors

Author name : Shweta Kharat
Education : Bachelor’s in Information Technology
Pursuing Masters in Information Technology