An Efficient Approach for Data Mining with Compressed Data

Mr.Vaibhav Kumar Sharma[1], Mr.Anil Gupta[2], Mr. B.L. Pal [3]
1 Student of M.Tech in CSE Mewar University, Chittorgarh, Rajasthan, India
2 Associate professor, MCA Deptt,M.R.S.C. Indore India
3 Asst. Professor, CSE Deptt,Mewar University, Chittorgarh, Rajasthan, India

Abstract
In an era of knowledge explosion, the growth of data increases rapidly day by day. Since data storage is a limited resource, how to reduce the data space in the process becomes a challenge issue. Data compression provides a good solution which can lower the required space. Data mining has many useful applications in recent years because it can help users discover interesting knowledge in large databases. Existing compression algorithms are not appropriate for data mining. In this paper our main focus is on association rule mining and data pre-process with data compression. We proposed a knowledge discovery process from compressed databases in which data can be decomposed.

Keywords: Association rule, Apriori Algorithm

I. INTRODUCTION

Data compression is one of good solutions to reduce data size that can save the time of discovering useful knowledge by using appropriate methods, for example, data mining. Data mining is used to help users discover interesting and useful knowledge more easily. It is more and more popular to apply the association rule mining in recent years because of its wide applications in many fields such as stock analysis, web log mining, medical diagnosis, customer market analysis, and bioinformatics. In this research, the main focus is on association rule mining and data pre-process with data compression. Proposed a knowledge discovery process from compressed databases in which can be decomposed into the following two steps:

.

A. Data pre-process step:

Data pre-process transforms the original database into a new data representation where several transactions are merged to become a new transaction. Eventually, it generates a new transaction database at the end of the data pre-process step.

B. Data mining step:

It uses an Apriori-like algorithm of association rule mining to find useful information. There are some problems in this approach. First, the compressed database is not reversible after the original database is transformed by the data pre-process step. It is very difficult to maintain this database in the future. Second, although some rules can be mined from the new transactions, it still needs to scan the database again to verify the result. This is because the data mining step produces potentially ambiguous results. It is a serious problem to scan the database multiple times because of the high cost of re-checking the frequent item sets.

It is even a bigger challenge to maintain the compressed database in the future. In addition, it spends too much time to check candidate item sets in the data mining step. In this research, a more efficient approach, called Mining Merged Transactions with the Quantification Table is proposed, which can compress the original database into a smaller one and perform the data mining process without the above problems

II. APRIORI ALGORITHM

A. Architecture of simple Apriori algorithm:

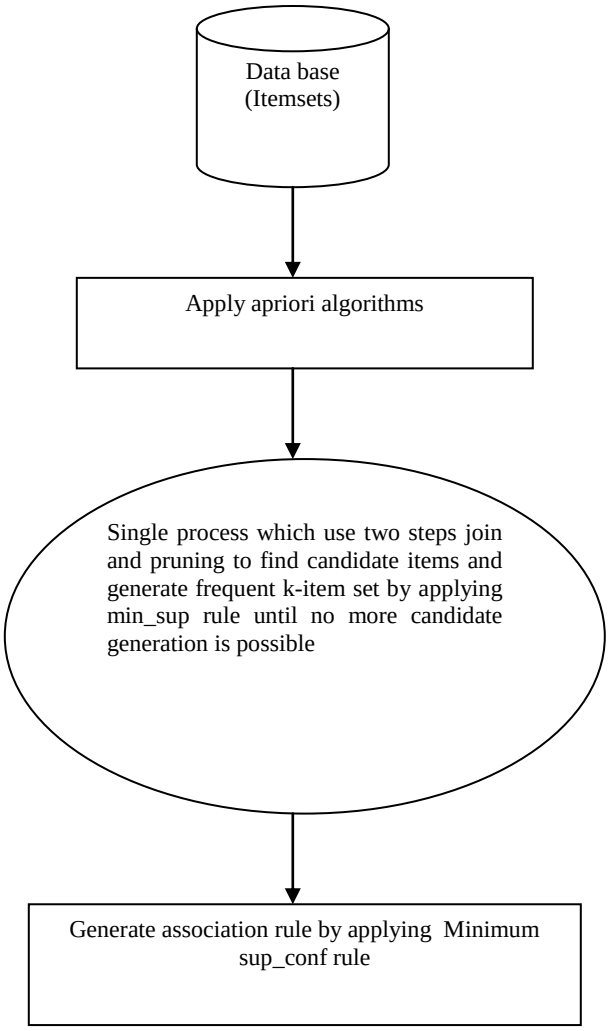

Figure 1: Architecture of simple Apriori algorithm

B. Flow Chart of simple Apriori Algoritm:

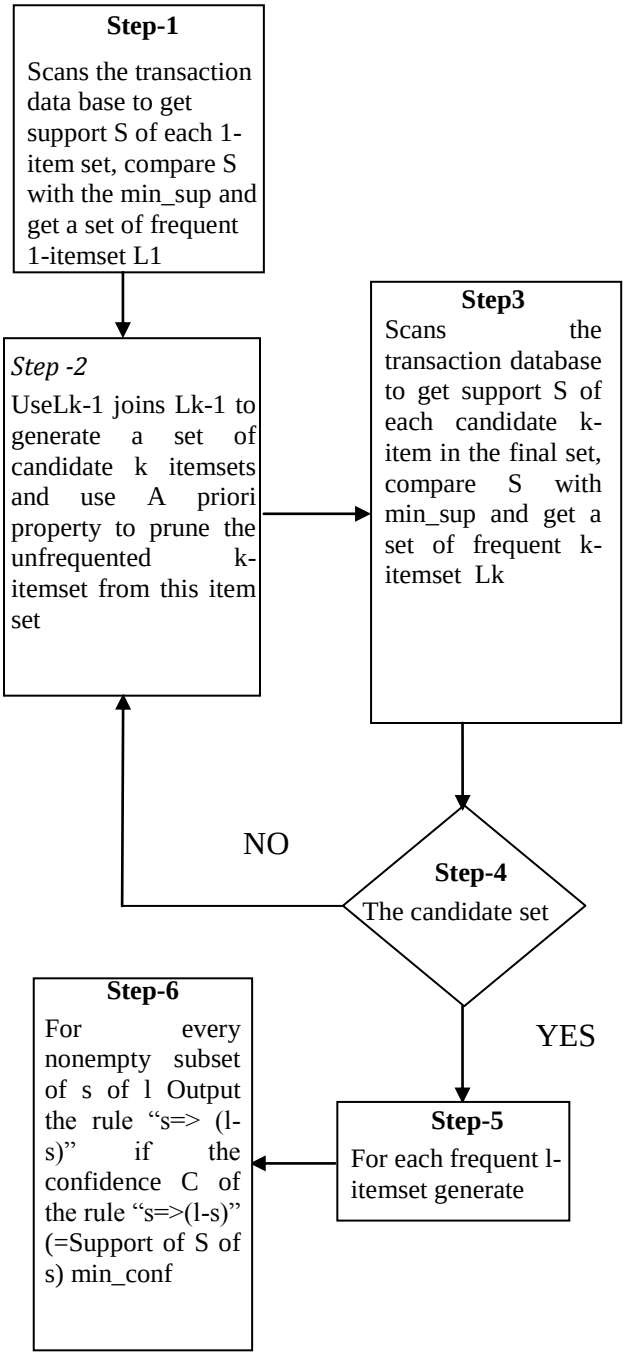

Figure 2: Flow Chart for Apriori Algorithm

C. Apriori Example:

TID	List of Items
T100	I1, I2, I5
T100	I2, I4
T100	I2, I3
T100	I1, I2, I4
T100	I1, I3
T100	I2, I3
T100	I1, I3
T100	I1, I2 ,I3, I5
T100	I1, I2, I3

D. **Pseudo code for Apriori Algorithm:**

- Join Step: Ck is generated by joining Lk-1with itself
- Prune Step: Any (k-1)-item set that is not frequent cannot be a subset of a frequent k-item set
- Pseudo-code:

Ck: Candidate item set of size k+1

Lk: frequent item set of size k+1

L1 = {frequent items};

for (*k*= 1; *Lk*!= ∅; *k*++) **do begin**

Ck+1 = candidates generated from *Lk+1*;

for each transaction *t* in database do

Increment the count of all candidates in *Ck+1* that are contained in *t*

Lk+1= candidates in *Ck+1*with min_support

end

III. LITERATURE REVIEW

Knowledge discovery process from compressed databases uses an Apriori-like algorithm of association rule mining to find useful information. There are some problems in this approach. First, the compressed database is not reversible after the original database is transformed by the data pre-process step. It is very difficult to maintain this database in the future. Second, although some rules can be mined from the new transactions, it still needs to scan the database again to verify the result. This is because the data mining step produces potentially ambiguous results. It is a serious problem to scan the database multiple times because of the high cost of re-checking the frequent item sets [1].

Cheng-Fa Tsoi. EChau Lin and Chi-Pin Chen, proposed an algorithm focuses on compressing related transactions and building a quantification table for pruning candidate item sets that are impossible to become frequent item sets. Finally, an example is provided to show the processes of our method. To simplify the description, it assumes the items in each transaction are presented in a lexicographical order. Algorithms like compress transactions to reduce the size of a transaction database. Then, they use Apriori-like algorithms to mine the compressed database [2].

Weimin Ouyang, identified three limitations in traditional algorithms for mining sequential patterns are built on the binary attributes databases. Firstly, it cannot concern quantitative attributes; secondly, only direct sequential patterns are discovered; thirdly, it cannot process these data items with similar frequencies which will result in the dilemma called the rare item problem. They put forward a discovery algorithm for mining both direct and indirect fuzzy sequential patterns with multiple minimum supports by combining these three extensions [3].

Pankaj Kumar Deva Sarma1 and Anjana Kakati Mahanta, find out three limitations in traditional algorithms for mining association rules are built on the binary attributes databases. Firstly, it cannot concern quantitative attributes; secondly, it

finds out frequent itemsets based on the single one user-specified minimum support threshold, which implicitly assumes that all items in the data have similar frequency; thirdly, only the direct association rules are discovered. Mining fuzzy association rules has been proposed to address the first limitation. In this paper, they put forward a discovery algorithm for mining both direct and indirect fuzzy association rules with multiple minimum supports to resolve these three limitations [4].

A research report on "A New Approach of Modified Transaction Reduction Algorithm for Mining" mentioned that the *Apriori* algorithm is a level wise search algorithm for mining frequent itemsets for Boolean association rules. The large itemsets are computed through iterations. In each iteration the database is scanned once and all large itemsets of same size are computed. The large itemsets are computed in the ascending order of their sizes. In the first iteration, the size 1- large itemsets are computed by scanning the database once. Subsequently, in the kth iteration *(k*> 1), a set of candidate sets Ck is created by applying the candidate set generating function Apriori-gen on *Lk* - 1, where *Lk* - 1 is the set of all large *(k-I)* itemsets found in the iteration *k-l.* Apriori-gen generates only those k-itemsets whose every *(k-l)*- itemsets subset is in *L* k-l' The support counts of the candidate itemsets in Ck are then computed by scanning the database once and the size-k large itemsets are extracted from the candidates [5].

The drawback of Apriori is that when the cardinality of the longest frequent itemsets is *k,* Apriori needs k passes of database scans. In addition, this algorithm is computation intensive in generating the candidate itemsets and computing the support values, especially for application with very low support threshold and / or vast amount of items. In this algorithm, if the number of first itemsets element is *k,* DB will be scanned *k* times at least. So it will have low efficiency. It is a base point to reduce the number of itemset for improving the Apriori algorithm [6].

Another research report on "New Coding Method to Reduce the Database Size and Algorithm with Significant Efficiency in Association Rule" Huimin HE, Haiyan DU, Yongjin LIU, Fangping LI, proposed a fast algorithm called Apriori in which generates (k+1)-candidates using joins over frequent k-itemset, all subset of one itemset must be generated by the algorithm. Although many of these frequent itemset may be not useful and may not exploit for finding association rules because some of these frequent itemset haven't any interestingness antecedent or consequent in rules but they had to generate them to find superior frequent itemset. Additionally size of database was main problem of this algorithm. Some modified algorithm of Apriori algorithm proposed to solve this problem but those algorithms also have the database size problem [7].

IV. PROBLEM DOMAIN

In an era of knowledge explosion, the growth of data increases rapidly day by day. Since data storage is a limited resource, how to reduce the data space in the process becomes a challenge issue. Data compression provides a good solution which can lower the required space. Data mining has many useful applications in recent years because it can help users discover interesting knowledge in large databases. Existing compression algorithms are not appropriate for data mining. In two different approaches were proposed to compress databases and then perform the data mining process. They all lack the ability to decompress the data to their original state and improve the data mining performance.

V. SOLUTION DOMAIN

The main focus of the proposed algorithm focuses will be on compressing related transactions and building a quantification table for pruning candidate item sets that are impossible to become frequent item sets. We will present a novel approach which can:

(a) Support local transaction variation

(b) Recover the transaction database to its original state

(c) Make the compressed database much smaller than the original one

(d) Reduce data mining time

Our approach "The Mining Merged Transactions with the Quantification Table" will has three phases:

(1) Merge related transactions to generate a compressed database

(2) Build a quantification table

(3) Discover frequent item sets

VI. PROPOSED ALGORITHM

Implementation of our Compressed Data Set Apriori Algorithm:

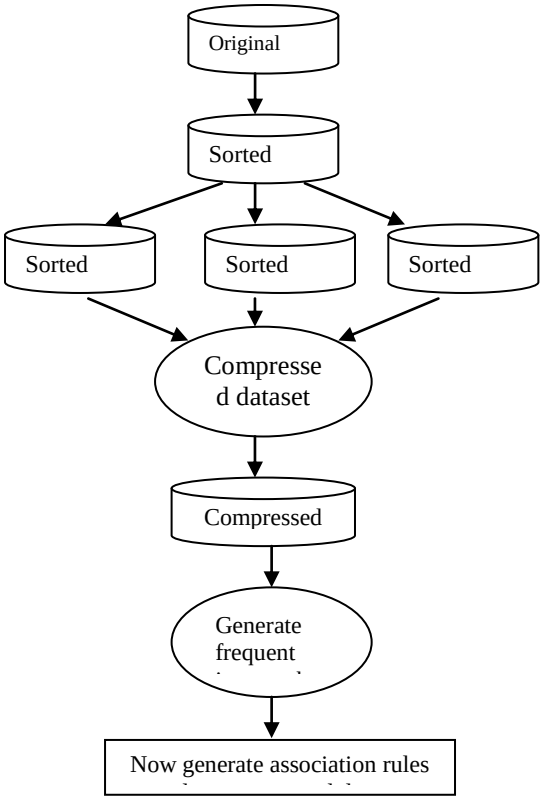

Figure 3: Architecture for compressed dataset Apriori

A. Pseudo code for our compressed dataset Apriori Algorithm:

Pseudo Code of Merge Mining Algorithm

// **Phase One**
Algorithm

1: Input: Database D and parameter K
2: Output: compressing patterns H
3: C ←− GetCandidate()
4: H ←− {∅}
5: while |H| < K do
6: for P ∈ C do
7: L∗(D|P) ←− CompressSize(D|P)
8: end for
9: P ←− argmin P (L(D|P))
10: H ←− H ∪ {P
}
11: C ←− C \ {P∗
}
12: Replace all instances of P∗by its pointers
13: end while
14: Return

Phase Two
Compress Size

Sub algorithm

1. D^M = Compressed database
2. L_1^M – {large1 – itemsets in compressed database};
3. for(k=2; $L_{k-1}^{M+1} \neq \Phi$; k++) do begin
4. C_k^M = merging – gen (L_{k-1}^M);
5. for all transactions $T^*_M \in D^{M+1}$ do begin

Algorithm

1: Input: Database D = {S1, S2, · · · , Sn} and pattern P
2: Output: L∗(D|P) optimal compressing size of Dgiven it is encoded by P
3: c ←− |P | + Σni=1|Si|
4: for Si ∈ D do
5: while Si has an instance of P does
6: s ←− the left-most instance of P in Si
7: Remove s from Si
8: c = c − |P | + 1
9: end while
10: end for
11: Return

Where:

c*	Candidate itemset in the merged database
l*	Large itemset in the merged database
L_K	Large k-itemset in the original transaction database
L^M_K	Large k-itemset in the merged database
C^M_K	Candidate k-itemset in the merged database
D^M	The merged database after preprocess
D	The original transaction database
T	A transaction in the original database
T*	A transaction in the merged database

T_N	All transaction in the original database
T^*_M	All transaction in the merged database
M+1	Number of groups after data prepocess

VII.CONCLUSION

Our experiments lead to the following conclusions:

(1) Form the proposed algorithm it is clear that compressed transaction data set methods is much better as compare to the other methods. Our proposed method compresses the original database into a smaller one and performs the data mining process efficiently. Our approaches have the following characteristics:

(a) The compressed database can be decompressed to the original form.

(b) Reduce the process time of association rule mining by using a quantification table.

(c) Reduce I/O time by using only the compressed database to do data mining.

(d) Allow incremental data mining.

References

1. Dr. M. Renuka Devi & Mrs. A. Baby sarojini, “Applications of Association Rule Mining in different Databases”. Published in Journal of Global Research in Computer Science REVIEW ARTICLE Volume 3, No. 8, August 2012, pp. 30-34.
2. Cheng-Fa Tsoi. EChau Lin and Chi-Pin Chen, “A new fast algorithms for mining association rules in large databases” Published in: Systems, Man and Cybernetics, 2002 IEEE International Conference on (Volume: 7) 6-9 Oct. 2002.
3. Weimin Ouyang, “Discovery of direct and indirect fuzzy sequential patterns with multiple minimum supports in transaction databases”. Published in Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference 29-31 May 2012, pp. 302 – 306.
4. Pankaj Kumar Deva Sarma1 and Anjana Kakati Mahanta, “Mining Direct and Indirect Fuzzy Association Rules with Multiple Minimum Supports in Large Transaction Databases”. Modern Education Technology Center, Shanghai University of Political Science and Law Shanghai 201701, China2011 IEEE.
5. Ramaraj Eswara Thevar & Rameshkumar Krishnamoorthy, “A New Approach of Modified Transaction Reduction Algorithm for Mining”. Alagappa University, Karaikudi 2008 IEEE.
6. Weimin Ouyang, Qinhua Huang, “Mining direct and indirect fuzzy association rules with multiple minimum supports in large transaction databases”. Published in Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on (Vol. 2) 26-28 July 2011, pp. 947-951.
7. Huimin HE, Haiyan DU, Yongjin LIU, Fangping LI, “New Coding Method to Reduce the Database Size and Algorithm with Significant Efficiency in Association Rul”. Yi XIE 2008 IEEE.
8. Jia -Yu Dai, Don – Lin Yang, Jungpin Wu and Ming-Chuan Hung, “ Data Mining Approach on Compressed Transactions Database” in PWASET Volume 30, pp 522-529,2010.
9. M. C. Hung, S. Q. Weng, J. Wu, and D. L. Yang, "Efficient Mining of Association Rules Using Merged Transactions," in WSEAS Transactions on Computers, Issue 5, Vol.5, pp. 916-923, 2008.
10. D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, and T. Yiu, "MAFIA: A maximal frequent itemset algorithm," IEEE Transactions on Knowledge and Data Engineering, Vol. 17, pp. 1490-1504, 2008.
11. D. Xin, J. Han, X. Yan, and H. Cheng, "Mining Compressed Frequent-Pattern Sets," in Proceedings of the 31st international conference on Very Large Data Bases, pp. 709-720, 2007.