Development of New Algorithm for Finding 2nd Level Association Rules using Fast Apriori Implementation

Arpna Shrivastava
Research Scholar
Barkat Ullah University, Bhopal

R.C. Jain
Director
SATI, Vidisha

Abstract

Need to explore data mining techniques for finding association rules are increasing every day due to increased market competition. Many algorithms based on Apriori presented to finding the association rules. Most of them present theoretical approach to improve the efficiency of Apriori algorithm. In this study, Fast implementation of Apriori algorithm modified to find out the association rules of second level. A new coding scheme also used for coding the customer transaction database. The coding of sample customer transaction database and the results are also discussed in this study.

Keywords

Association rules, Apriori algorithm, support, confidence, multilevel association rules.

1. INTRODUCTION

Association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. With wide applications of computers and automated data collection tools, massive amounts of transaction data have been collected and stored in databases. Discovery of interesting association relationships among huge amounts of data will help marketing, decision making, and business management. Therefore, mining association rules from large data sets has been a focused topic in recent research into knowledge discovery in databases [1]. Apriori is a classic algorithm for mining frequent item sets and learning association rules of single level [2].

Mining multilevel association rule was first introduced in [3]. Multilevel association rules provide more specific and concrete knowledge. Apriori based algorithm for multiple-level association rules from large database was presented in [5].

Mining association from numeric data using genetic algorithm is explored and the problems faced during the exploration are discussed in [13]. Positive and negative association rules are another aspect of association rule mining. Context based positive and negative spatio-temporal association rule mining algorithm based on Apriori algorithm is discussed in [14]. Association rule generation requires scan of the whole databases which is difficult for very large database. An algorithm for generating Samples from large databases is discussed in [11]. An improved algorithm based on Apriori algorithm to simulate car crash is discussed in [12].

There are many algorithms presented which are based on Apriori algorithm [4,6,7,9]. The efficiency of algorithms is based on their implementation. UML class diagram of Apriori algorithm and its Java implementation is presented in [10]. A fast implementation of Apriori algorithm was presented in [8]. The central data structure used for the implementation was Trie because it outperforms the other data structure i.e. Hash tree.

In this paper, a new coding scheme is developed to code the transaction table. The coding scheme is used by the modified fast implementation of Apriori algorithm for finding the first and second level association rules. The new algorithm is more efficient and effective.

2. INPUT CODING

The transaction data is coded into pre-specified codes. Initially, 100 categories of items are taking into consideration and every item has 100 brands. For example, milk is represented by 10 and bread is represented by 11. And the brand of milk is represented by 12 for Amul. So Amul milk is represented by 1012. This coding scheme can be extended further for 3rd level and so on. This coding scheme can produce maximum of 100×100 matrix as input file.

The table 2.1 and table 2.2 show the coding scheme for items and their category respectively.

Table 2.1: Coding scheme for items

S.No.	Item	Code
1	Milk	10
2	Bread	11
3	Biscuit	12
4	Butter	13
5	Atta	14

Table 2.2: Coding scheme for categories of milk

S.No.	Item	Code
1	Amul	20
2	Mother Dairy	21
3	Sanchi	22
4	Paras	23
5	Jersey	24

The table 2.3 shows the complete code for milk with its brand.

Table 2.3: Coding scheme for milk with brands

S.No.	Item with brand	Code
1	Amul Milk	1020
2	Mother dairy milk	1021
3	Sanchi Milk	1022
4	Paras Milk	1023
5	Jersey Milk	1024

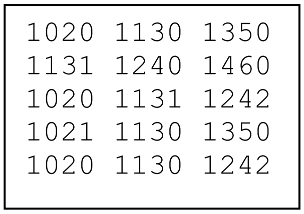
Similarly others items are also coded from the transaction table. This scheme gives the complete code for item with their brands.

Five transactions are taken as Sample transactions from super market database.

Table 2.4: Transaction table

Tid	Item purchased
1	{Milk(Amul), Bread(Harvest), Atta(Ashirvad)}
2	{Bread(Britania), Biscuit(Britania), Noodles(Maggi)}
3	{Milk(Amul), Bread(Britania), Biscuit(Parle)}
4	{Milk(Mother Dairy), Bread(Harvest), Atta(Ashirvad)}
5	{Milk(Amul), Bread(Harvest), Biscuit(Parle)}

Transaction Ids and items purchased against each transaction id are shown in the table 2.4. Input file Data.dat prepared using the coding scheme for transaction database. The input file is shown in fig.2.1.



```
1020 1130 1350
1131 1240 1460
1020 1131 1242
1021 1130 1350
1020 1130 1242
```

Fig 2.1: Data.dat file

Each row of input file is representing one transaction of the database.

3. MODIFIED ALGORITHM

Apriori algorithm is a classic algorithm for finding frequent item sets and single level association rules [4]. A fast implementation of Apriori algorithm is presented using the trie data structure in [8]. Bodon implementation generates frequent item sets and association rules of single level. It does not generate the association rules of second level.

This Bodon implementation has been modified for finding the association rules of second level. To facilitate the process of finding the level of association rules one argument has been added. The necessary modifications are also done to process this new argument. One additional function is added to separate the code of input file. After separating the coded inputs, it calls the function to generate the association rules according to their required level. This new addition of code is shown as follows

```
Loop till character is greater or equal to '0' and character is less than equal to '9'.
( i )  split or extract the positional character of file pointer.
( ii )  add the current character with previously produced value in step 1 (i).
( iii)  get character of incremented index.
( iv )  increment the position.
If level = 1 then
( a ) converting int [ item ] to char [ items ] i.e. Casting item ( int type ) to items (char).
( b ) extracting the product digits from item.
Else if level = 2 then
( a ) converting int [ item ] to char [ item ] ie. Casting item ( int ) type to items (char).
( b ) extr acting the brands digits ( code ) from item.
If produced or brand code is NULL or ZERO then
          Item = 0
If position isnot NULL or ZERO then
Insert item into temporary basket.
```

In step 1, it identifies the codes and separates them. If required level is one then it separates the item codes from their categories code and calls the association rule generation function. Else it calls the association rule generation function on both items and their categories. The results are stored in the file named out.txt which is passed as argument to the program.

4. RESULTS

The results are generated for both levels of association rules and the frequent item sets. The frequent item set for level 1 association rules are given in table 4.1.

Table 4.1: Frequent 1-Item sets

S. No.	Item code (occurrence)	Item Name (occurrence)
1	14 (1)	Noodles (1)
2	13 (2)	Atta (2)
3	12 (3)	Biscuit(3)
4	10 (4)	Milk (4)
5	11 (5)	Bread (5)

Similarly frequent 2-itemsets and frequent 3-itemsets are also found. They are shown in table 4.2 and table 4.3 respectively.

Table 4.2: Frequent 2-Itemsets

S. No.	Item code (occurrence)	Item Name (occurrence)
1	14 12 (1)	Noodles, Biscuit (1)
2	14 11 (1)	Noodles, Bread (1)
3	13 10 (2)	Atta, Milk (2)
4	13 11 (2)	Atta, Bread (2)
5	12 10 (2)	Biscuit, Milk (2)
6	12 11 (3)	Biscuit, Bread (3)
7	10 11 (4)	Milk, Bread (4)

Table 4.3: Frequent 3-Itemsets

S. No.	Item code (occurrence)	Item Name (occurrence)
1	14 12 11 (1)	Noodles, Biscuit, bread (1)
2	13 10 11 (2)	Atta, Milk, Bread (2)
3	12 10 11 (2)	Biscuit, Milk, Bread (2)

Frequent 4-itemsets are not provided in the input file so they are not generated by this algorithm. Association rules generated by the algorithm in out.txt are given with item names in table 4.4

Table 4.4: Association rules of level-1

Item Code ➔Item Code (confidence, occurrence)	Item Name ➔ Item Name (Confidence, Occurrence)
14 ==> 12 (1, 1)	Noodles➔Biscuit (1,1)
14 ==> 12 11 (1, 1)	Noodles ➔Biscuit Bread (1,1)
14 ==> 11 (1, 1)	Noodles➔ Bread (1,1)
13 ==> 10 (1,2)	Atta➔Milk (1,2)
13 ==> 10 11 (1, 2)	Atta➔ Milk Bread (1,2)
13 ==> 11 (1, 2)	Atta➔ Bread (1,2)
12 ==> 10 (0.67,2)	Biscuit➔ Milk (0.67,2)
12 ==> 10 11 (0.67,2)	Biscuit➔ Milk Bread (0.67,2)
12 ==> 11 (1,3)	Biscuit➔ Bread (1,3)
11 ==> 12 (0.6,3)	Bread➔ Biscuit (0.6,3)
10 ==> 11 (1,4)	Milk➔ Bread (1,4)
11 ==> 10 (0.8,4)	Bread➔Milk (0.8,4)

These association rules are generated for minimum support value of 0.05 and minimum confidence value of 0.5 for level 1. Similarly level-2 association rules and frequent item sets are also generated. Frequent item sets are shown in table 4.5.

Table 4.5: Frequent 1-Itemsets

S. No.	Item code (occurrence)	Item Name-Brand (occurrence)
1	1460 (1)	Noodles-Maggi(1)
2	1240 (1)	Biscuit-Britania (1)
3	1021 (1)	Milk-Mother Dairy (1)
4	1350 (2)	Atta-Ashirvad (2)
5	1242 (2)	Biscuit-Parle (2)
6	1131 (2)	Bread-Britania (2)
7	1130 (3)	Bread-Harvest (3)
8	1020 (3)	Milk-Amul (3)

Similarly frequent 2-item sets and frequent 3-item sets can be given. The association rules for level-2 for minimum support value of 0.05 and minimum confidence value of 0.5 are generated and shown in table 4.6.

Table 4.6: Association rules of level-2

Item Code ➔Item Code (confidence, occurrence)	Item Name ➔ Item Name (Confidence, Occurrence)
1460 ==> 1240 (1,1)	Noodles-Maggi ➔Biscuit-Britania (1,1)
1240 ==> 1460 (1,1)	Biscuit-Britania ➔ Noodles-Maggi (1,1)
1460 ==> 1240 1131(1,1)	Noodles-Maggi ➔ Biscuit-Britania, Bread-Britania (1,1)
1240 ==> 1460 1131 (1,1)	Biscuit-Britania ➔ Noodles-Maggi, Bread-Britania (1,1)
1460 ==> 1131 (1,1)	Noodles-Maggi ➔ Bread-Britania (1,1)
1240 ==> 1131 (1,1)	Biscuit-Britania ➔ Bread-Britania (1,1)
1021 ==> 1350 (1,1)	Milk-Mother Dairy➔Atta-Ashirvad Milk (1,1)
1021 ==> 1350 1130 (1,1)	Milk-Mother Dairy ➔ Atta-Ashirvad (1,1)
1021 ==> 1130 (1,1)	Milk-Mother Dairy ➔ Bread-Harvest (1,1)
1350 ==> 1130 (1,2)	Atta-Ashirvad➔ Bread-Harvest (1,2)
1130 ==> 1350 (0.67,2)	Bread-Harvest➔ Atta-Ashirvad (0.67,2)
1350 1020 ==> 1130 (1,1)	Atta-Ashirvad,Milk-Amul ➔Bread-Havest (1,1)
1242 1131 ==> 1020	Biscuit-Parle➔Milk-Amul (1,1)

(1,1)	
1131 1020 ==> 1242 (1,1)	Bread-Britania, Milk-Amul➔ Biscuit-Parle (1,1)
1242 1130 ==> 1020 (1,1)	Biscuit-Parle, Bread-Harvest➔Milk-Amul (1,1)
1242 ==> 1020 (1,2)	Biscuit-parle➔Milk-Amul (1,2)
1020 ==> 1242 (0.67,2)	Milk-Amul➔ Biscuit-Parle (0.67,2)
1130 ==> 1020 (0.67,2)	Bread-Harvest ➔Milk-Amul (0.67,2)
1020 ==> 1130 (0.67,2)	Milk-Amul➔Bread-Harvest (0.67,2)

Association rules of level-2 are found which shows that implementation by modifying the fast implementation of Apriori is successful and the results are satisfactory.

5. CONCLUSION

Fast implementation of Apriori algorithm analyzed and modified it to find frequent item sets and association rules of level-2. The modification is done in two steps. In first step, the transaction database is coded using a new coding scheme and in second step the code of implementation modified and a new module is added to facilitate the second level association rule generation.

This algorithm is based on fast implementation of Apriori algorithm hence it is more efficient. It can be further enhanced to level-3 association rules and so on.

6. REFERENCES

[1] R. Agrawal, T. Imielinski; A. Swami: Mining Association Rules Between Sets of Items in Large Databases", SIGMOD Conference 1993, pp. 207-216.

[2] R. Agrawal, R. Srikant, “Fast Algorithms for Mining Association Rules”, Proceedings of the 20th International Conference on Very Large Data Bases, 1994, pp. 487-499.

[3] J. Han, Y. Fu, “Discovery of Multiple-Level Association Rules from Large Database”, Proceeding of the 21st VLDB Conference Zurich, Swizerland, 1995, pp.420-431.

[4] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In Advances in Knowledge Discovery and Data Mining, 1996, pp. 307.328.

[5] J. Han, Y. Fu, “Mining Multiple-Level Association Rules in Large Database”, IEEE transactions on knowledge & data engineering in 1999, pp.1-12.

[6] Bing Liu,Wynne Hsu and Yiming Ma, “Mining association rules with multiple minimum supports”, ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1999, pp.337-341.

[7] F. Berzal, J. C. Cubero, Nicolas Marin, and Jose-Maria Serrano, “TBAR: An efficient method for association rule mining in relational databases”, Data and Knowledge Engineering 37, 2001, pp.47-64.

[8] F. Bodon, “Fast Apriori Implementation”, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, 2003.

[9] N. Rajkumar, M.R. Kartthik and S.N. Sivanandam, “Fast Algorithm for Mining Multilevel Association Rules”, Conference on Convergent Technologies for the Asia-Pacific Region, TENCON, 2003, pp.688-692.

[10] Y. Li, “The Java Implementation of Apriori algorithm Based on Agile Design Principles”, 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), 2010, pp. 329 – 331.

[11] B. Chandra, S. Bhaskar, “A new approach for generating efficient sample from market basket data”, Expert Systems with Applications (38), Elsevier, 2011, pp. 1321–1325.

[12] L. Xiang, “Simulation System of Car Crash Test in C-NCAP Analysis Based on an Improved Apriori Algorithm”, International Conference on Solid State Devices and Materials Science, Physics Procedia (25), Elsevier, 2012, pp. 2066 – 2071.

[13] B. Minaei-Bidgoli, R. Barmaki, M. Nasiri, “Mining numerical association rules via multi-objective genetic algorithms”, Information Sciences (233), Elsevier, 2013, pp.15–24.

[14] M. Shaheen, M. Shahbaz, A. Guergachi, “Context based positive and negative spatio-temporal association rule mining”, Knowledge-Based Systems (37), Elsevier, 2013, pp. 261–273.