British Journal of Mathematics & Computer Science
4(2): 241-251, 2014

SCIENCEDOMAIN *international*
www.sciencedomain.org

Generating 3rd Level Association Rules Using Fast Apriori Implementation

Arpna Shrivastava[1*], R. C. Jain[2] and A. K. Shrivastava[3]

[1]*Research Scholar, SATI, Vidisha, India.*
[2]*SATI, Vidisha, India.*
[3]*Department of CA, KIET, Ghaziabad, India.*

Original Research Article

Received: 28 June 2013
Accepted: 03 October 2013
Published: 23 October 2013

Abstract

The mining of association rules and frequent item sets are the main area of interest in recent research activities. The multiple level association rules provide the more meaningful information in comparison to single level association rules which describes only single level concept hierarchy. The Apriori algorithm is most established algorithm for mining the single level association rules. In this study, the fast implementation of Apriori algorithm has been used to generate 3rd level association rules with some modifications in the algorithm. The data coding and data cleaning techniques are used to find the multilevel association rules as they are prerequisite to implement the modified algorithm.

Keywords: *Multiple-level association rule, fast Apriori implementation, support, confidence, recoding, data cleaning.*

1 Introduction

The data mining will be most useful in future because we are storing huge amount of data per day. The new information can be explored by applying the data mining techniques on this huge amount of stored data. The mining of association rule is most useful technique for data mining. The multilevel association rules provide the information at different level of concept hierarchy [1].

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. Piatetsky-Shapiro et al. [2] describes analyzing and presenting strong rules discovered in databases using different measures of interestingness. Based on the concept of strong rules, Agrawal et al. [3] introduced association rules for discovering regularities between products in large scale transaction data recorded by point-of-sale (POS) systems in supermarkets.

*Corresponding author: arpna.10878@gmail.com;

The Apriori algorithm is a single-level association-rule mining algorithm. By single level, we mean to say that there are no hierarchies among items in an itemset. In contrast, in multiple-level association rule mining, the items in an itemset are characterized by using a concept hierarchy. Mining occurs at multiple levels in the hierarchy. At lowest levels, it might be that no rules may match the constraints. At highest levels, rules can be extremely general. Generally, a top-down approach is used where the support threshold varies from level to level [4]. In many of the applications of single-level association rule mining, the items contained in an itemset could potentially be hierarchically organized where primitive level concepts can be generalized to higher levels while the more specific items to lower levels of the hierarchy. By forming such a concept hierarchy, a process of discovering association rules at multiple concept levels progressively deepens the knowledge mining process for finding more meaningful and refined knowledge from the data [1,4].

Mining association from numeric data using genetic algorithm is explored and the problems faced during the exploration are discussed in [5]. 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 [6]. 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 [7]. An improved algorithm based on Apriori algorithm to simulate car crash is discussed in [8]. There are many algorithms presented which are based on Apriori algorithm [9, 10,11,12]. The efficiency of algorithms is based on their implementation. UML class diagram of Apriori algorithm and its Java implementation is presented in [13]. A fast implementation of Apriori algorithm was presented in [14]. The central data structure used for the implementation was Trie because it outperforms the other data structure i.e. Hash tree.

Apriori algorithm is the best algorithm for single level association rules mining [3,15]. In this study, the generation of 3rd level association rules using fast Apriori implementation has been discussed. Implementation of this modified algorithm for generating 3rd level association rules requires the repeated call of this algorithm. Database should be coded in specified format and data cleaning has been done if required, generation of frequent item sets and in the last generation of 3rd level association rules.

This paper is organised as follows. Problem is defined in section 2. Section 3 discusses the coding of transaction database. Section 4 deals with cleaning of data if required. Modified algorithm is presented in section 5 Results are discussed in section 6. Section 7 deals with conclusion and future scope.

2 Problem Statement

The items in any super market are numbered using the barcode. It facilitated the automatic reading of item details using the barcode reader. Barcode for an item can be some logical code or just a sequence number. The transaction database of any super market contains the transaction id and the set of barcodes against each transaction id. The sample transaction table is shown in Table 1.

Table 1. Transactional database

Transaction id	Barcodes
12345	{121234, 102302, 876546}
12346	{121212, 102302, 121234}
	……………………………..

The item master table contains the details of item against the each barcode. If barcode is just a sequence number then mapping of item details from item master database to transaction database is required to produce some meaningful database. Table 2 shows the sample item master database.

Table 2. Item master database

Barcode	Category	Brand	Pack	Price (Rs.)
121234	Bread	Harvest	Normal	18
121212	Milk	Amul	500ml	22
……	….	……..	……..	……….

Item master database contains the complete details of items against each barcode. The barcode 121234 represents the item category bread brand harvest, pack normal and price Rs.18/-. This item master is providing three level of concept hierarchy. First level the item category, on the second level the item brand and the third level is pack. By 3rd level association rules, the association between normal pack harvest bread with 500ml amul milk will be explored. First the frequent item sets are explored and then the association rules are explored. The support and confidence are different for every level threshold value and they are user defined. The different threshold for every level will produce good number of association rules.

3 Coding of Data

The algorithm runs of coded database. In this study, the six digits code has been used for every item purchased. The six digits of the code has been divided into three level hierarchy so two digits per level. In this study, maximum hundred categories can be coded. Every category of item can have maximum of hundred brands and every brand for given category can have maximum of hundred packing options. This coding can be flexible in future studies. Using three tables of code and item category, brand and packs, the coding of the database has been done easily.

Sample coding scheme is shown in Table 3. Every item category is represented by two digit code. By this approach, maximum of hundred item category can be coded. So after reading the item category the program which is responsible to generate the codes will generate two digits code for every item category.

Table 3. Coding scheme for item categories

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

Table 4 is showing the sample coding scheme for brand of item category of milk. So every item category can have hundred brands. The program which is responsible to generate the code will put two digits code for the brand name of item category. For example for brand amul of item category milk, the code is 20.

The Table 5 is showing the sample code for packing options of items. Generally item comes in various packing options. By this coding scheme, maximum of hundred packing option are available to code the packing options for every brand of item category. For example 200ml pack of brand amul of item category milk 102000.

Table 4. Coding scheme for brands of item category milk

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

Table 5. Coding scheme for packs of brand amul item category milk

S.No.	Item pack (ml)	Code
1	200	00
2	500	01
3	1000	02
4	2000	03

The complete coding scheme of items is shown in Table 6. The program will generate six digits code for every item purchased. For example, 102101 is the code for item category milk, item brand mother dairy and packing of 500ml. The results will come in the form of frequent item sets and association rules of 3rd level and decoded easily using these three tables.

Table 6. Coding scheme for milk with brands

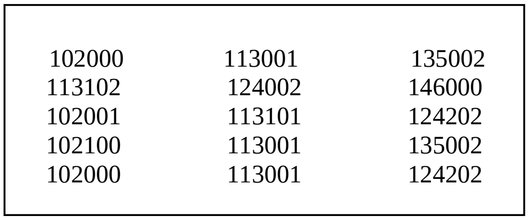
S.No.	Item with brand	Code
1	Amul Milk 200ml	102000
2	Mother dairy milk 500ml	102101
3	Sanchi Milk 200ml	102200
4	Paras Milk 1000ml	102302
5	Jersey Milk 2000ml	102403

Table 7 is displaying the sample transaction table of any super market. Transaction id is assigned against each purchase from the store. For example the first customer purchases the milk of amul brand in 200ml pack, bread of harvest brand in normal pack and ashirvad atta in 2 kg pack.

Table 7. Transaction table

Tid	Item purchased
1	{Milk(Amul (200ml)), Bread(Harvest(normal)), Atta(Ashirvad(2 kg))}
2	{Bread(Britania(big pack)), Biscuit(Britania(100gm)), Noodles(Maggi(small))}
3	{Milk(Amul(500ml)), Bread(Britania(normal)), Biscuit(Parle(100gm))}
4	{Milk(Mother Dairy(200ml)), Bread(Harvest(normal)), Atta(Ashirvad(2 kg))}
5	{Milk(Amul(200ml)), Bread(Harvest(normal)), Biscuit(Parle(100gm))}

The program which is responsible to code the database will generate the data.dat file for the coded database. Each row of this file contains the one row of transaction table. The row number will represent the transaction id and the contents of the row will represent the item purchased against that transaction id. The sample of data.dat file is shown in Fig. 1.



```
102000          113001          135002
113102          124002          146000
102001          113101          124202
102100          113001          135002
102000          113001          124202
```

Fig. 1. Data.dat file

This is the input file for the modified Apriori algorithm. The implementation of this modified algorithm will produce the frequent item sets and then the association rules of 3rd level.

4 Cleaning of Data

Cleaning of data is required for the databases which are already available in coded form. Another program has been developed for cleaning the data files. The program takes the .dat file as input and done the cleaning process. It fills the missing digits by copying the digits are available within the code. After the cleaning process, it generates the new .dat file which has all six digit codes.

The program reads the every code from data.dat file and counts the digits of the code. If code is less than 6 digits it makes the code of six digits by adding the missing digits from the code.

The algorithm of data cleaning is given in Fig. 2. The algorithm of data cleaning takes the data.dat file as input. It opens in this input file in read mode, creates another out.dat file and opens this out.dat file in write mode. It reads the data.dat file and checks for space and new line character. These characters are the separators between two codes. It writes data into out.dat from data.dat by making all codes of six digits. At the end, it renames out.dat file to data.dat file.

This algorithm return data.dat file which have all six digits code into it. It completes our data cleaning process. It is a complete input file so our algorithm for finding frequent item sets and association rules will work properly.

```
Algorithm Data_Cleaning(data.dat)
{
        Open data.dat file in read mode
        Create and open out.dat in write mode
        i=0
                While( data.dat)
                {
                read data.dat into x
                if x=’  ‘ or x= new line then
                        if i=1 then
                        for j =1 to 5 do
                        item[j]=item[j-1]
                        if i=2 then
                        for j =2 to 5 do
                        item[j]=item[j-2]
                        for j=0 to 5-I do
                        write into out.dat
                else
                        write into out.dat
                        i=i+1
                }
Rename (data.dat, out.dat)
}
```

Fig. 2. Algorithm of data cleaning

5 Algorithm

The Apriori algorithm is a classic algorithm for finding frequent item sets and single level association rules [7]. A fast implementation of Apriori algorithm is presented using the trie data structure in [6]. 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 named level of association rule has been added. One additional function is added to separate the code of input file. After separating the coded inputs, it calls the main Apriori function to generate the association rules. This new addition of code is shown in Fig. 3.

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.

```
Modified_Apriori(input file, output file, level, min_support)
{
Swith ( level)
If level = 1
Then Apriori(Func_Separation(input file, level), output file, min_support, min_conf)
If level =2
Then Apriori(Func_Separation(input file, level), output file, min_support, min_conf)
Else
Apriori(Func_Separation(input file, level), output file, min_support, min_conf)
}

Func_Separation(input file, level)
{
Item=first element read from file
If level == 1
Then
String item = convert digit to string
String sub = item.substr(0,2)
If sub ==item
Extracted_item=sub
Then exit
Else
If level ==2
Sub= item.substr(0,4)
If sub== item
Extracted_item=sub
Then exit
Else
If level==3
Extracted_item=sub
}
```

Fig. 3. Modified code

6 Results

The results are generated for all levels of association rules and the frequent item sets. The frequent item set for level 3 association rules are given in Table 8.

Table 8. Frequent 1-Item sets

S. No.	Item code (occurrence)	Item Name (occurrence)
1	146000 (1)	Noodles(Maggi(small)) (1)
2	124002 (1)	Biscuit(Britania(100gm)) (1)
3	113102 (1)	Bread(Britania(big pack)) (1)
4	113101 (1)	Bread(Britania(normal)) (1)
5	102100 (1)	Milk(Mother Dairy(200ml)) (1)
6	102001 (1)	Milk(Amul(500ml)) (1)
7	135002 (2)	Atta(Ashirvad(2 kg)) (2)
8	124202 (2)	Biscuit(Parle(100gm)) (2)
9	102000 (2)	Milk(Amul(200ml)) (2)
10	113001 (3)	Bread(Harvest(normal)) (3)

Similarly frequent 2-itemsets and frequent 3-itemsets are also found. They are shown in Table 9 and Table 10 respectively.

Table 9. Frequent 2-Itemsets

S. No.	Item code (occurrence)	Item Name (occurrence)
1	146000 124002 (1)	Noodles(Maggi(small)) Biscuit(Britania(100gm)) (1)
2	146000 113102 (1)	Noodles(Maggi(small)) Bread(Britania(big pack)) (1)
3	124002 113102 (1)	Biscuit(Britania(100gm))Bread(Britania(big pack))(1)
4	113101 102001 (1)	Bread(Britania(normal)) Milk(Amul(500ml)) (1)
5	113101 124202 (1)	Bread(Britania(normal)) Biscuit(Parle(100gm)) (1)
6	102100 135002 (1)	Milk(Mother Dairy(200ml)) Atta(Ashirvad(2kg)) (1)
7	102100 113001 (1)	Milk(MotherDairy(200ml)) Bread(Harvest(normal))(1)
8	102001 124202 (1)	Milk(Amul(500ml)) Biscuit(Parle(100gm)) (1)
9	135002 102000 (1)	Atta(Ashirvad(2 kg)) Milk(Amul(200ml)) (1)
10	135002 113001 (2)	Atta(Ashirvad(2 kg)) Bread(Harvest(normal))(2)
11	124202 102000 (1)	Biscuit(Britania(100gm)) Milk(Amul(200ml)) (1)
12	124202 113001 (1)	Biscuit(Britania(100gm)) Bread(Harvest(normal))(1)
13	102000 113001 (2)	Milk(Amul(200ml)) Bread(Harvest(normal))(2)

Table 10. Frequent 3-Itemsets

S. No.	Item code (occurrence)	Item Name (occurrence)
1	146000 124002 113102 (1)	Noodles(Maggi(small)) Biscuit(Britania(100gm)) Bread(Britania(big pack))(1)
2	113101 102001 124202 (1)	Bread(Britania(normal)) Milk(Amul(500ml)) Biscuit(Britania(100gm))(1)
3	102100 135002 113001 (1)	Milk(Mother Dairy(200ml)) Atta(Ashirvad(2kg)) Bread(Harvest(normal)) (1)
4	135002 102000 113001 (1)	Atta(Ashirvad(2 kg)) Milk(Amul(200ml)) Bread(Harvest(normal))(1)
5	124202 102000 113001 (1)	Biscuit(Britania(100gm)) Milk(Amul(200ml)) Bread(Harvest(normal))(1)

The input file did not contain any transactions having four item purchased together. So it is not possible to generate the frequent 4-itemsets. Hence they are not generated by this algorithm.

Association rules generated by the algorithm in out.txt are given Fig. 4.

```
Association rules:
condition ==> consequence (confidence, occurrence)

146000 ==> 124002 (1, 1)
124002 ==> 146000 (1, 1)
146000 ==> 124002 113102 (1, 1)
124002 ==> 146000 113102 (1, 1)
113102 ==> 146000 124002 (1, 1)
146000 ==> 113102 (1, 1)
113102 ==> 146000 (1, 1)
124002 ==> 113102 (1, 1)
113102 ==> 124002 (1, 1)
113101 ==> 102001 (1, 1)
102001 ==> 113101 (1, 1)
113101 ==> 102001 124202 (1, 1)
102001 ==> 113101 124202 (1, 1)
113101 ==> 124202 (1, 1)
102100 ==> 135002 (1, 1)
102100 ==> 135002 113001 (1, 1)
102100 ==> 113001 (1, 1)
102001 ==> 124202 (1, 1)
135002 ==> 113001 (1, 2)
113001 ==> 135002 (0.666667, 2)
102000 ==> 113001 (1, 2)
113001 ==> 102000 (0.666667, 2)
```

Fig. 4. 3rd Level association rules

7 Conclusions and Future Scope

Fast implementation of Apriori algorithm analyzed and modified it to find frequent item sets and association rules of level-3. The modification is done in three steps. In first step, the transaction database is coded using a new coding scheme and in second step the cleaning of database is done if required and at the third step code of implementation modified and a new module is added to facilitate the third level association rules generation.

This algorithm is based on fast implementation of Apriori algorithm and generating the third level of association rules. This algorithm can be enhanced to extract fourth level association rules by providing the transaction data using the concept hierarchy of fourth level. Similarly it can be further enhanced to extract association rules of level-n.

Competing interests

Authors have declared that no competing interests exist.

References

[1] Jiawei H, Yongjian F. Discovery of multiple-level association rules from large database Proceeding of the 21st VLDB Conference Zurich, Swizerland. 1995;420-431.

[2] David WC, Vincent TN, Benjamin WT. Maintenance of discovered knowledge: A case in multi-level association rules Proceeding of 2nd international conference on knowledge discovery and data mining. 1996;1-10.

[3] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. SIGMOD Conference. 1993;207-216.

[4] Bing LWH, Yiming M. Mining association rules with multiple minimum supports KDD-99 San Diego CA USA. 1999;337-341.

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

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

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

[8] 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;2066 – 2071.

[9] Agrawal R, Mannila H, Srikant R, Toivonen H, Verkamo AI. Fast discovery of association rules In Advances in Knowledge Discovery and Data Mining. 1996;307-328.

[10] Bing LWH, Yiming M. Mining association rules with multiple minimum supports ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1999;337-341.

[11] Berzal F, Cubero JC, Nicolas M, Jose MS. TBAR: An efficient method for association rule mining in relational databases Data and Knowledge Engineering 37. 2001;47-64.

[12] Rajkumar N, Kartthik MR, Sivanandam SN. Fast algorithm for mining multilevel association rules Conference on Convergent Technologies for the Asia-Pacific Region, TENCON. 2003;688-692.

[13] Li Y. The java implementation of apriori algorithm based on agile design principles 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT). 2010;329 – 331.

[14] Bodon F. Fast apriori implementation Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations. 2003; 90.

[15] Agrawal R, Srikant R. Fast algorithms for mining association rules In Proceedings of VLDB. 1994;487-499.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=303&id=6&aid=2355