

Research Paper

**Generating 3rd Level Association Rules Using
Fast Apriori Implementation**

ABSTRACT

Frequent item sets and association rules are the main area of interest of any data mining activity. Multiple level association rules provide the meaningful information like Harvest brown bread is purchased with Amul full cream milk. Apriori algorithm is most established algorithm for single level association rule mining. Fast implementation of Apriori algorithm has been used to generate 3rd level association rules with required modifications in the algorithm. Data coding and data cleaning has been done which are prerequisite to this modified algorithm implementation.

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

1. INTRODUCTION

Data mining will be most useful in future because we will get new and useful information with the use of existing data. Finding useful information using some techniques and algorithms from large and huge amount of data is called 'data mining' [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, 4] 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, 5] introduced association rules for discovering regularities between products in large scale transaction data recorded by point-of-sale (POS) systems in supermarkets.

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, 2, 3].

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 [6]. 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 [1, 6]. 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, 6]. However, previous work has been focused on mining association rules at a single concept level. There are applications which need to find at multiple concept levels. For example, besides finding 80% of customers that purchase dal (Lentil) may also purchase grain, it could be informative to also show that 75% of people buy wheat if they buy chana dal (Split Bengal Gram). The association relationship in the latter statement is expressed at a lower concept level but often carries more specific and concrete information than that in the former. This requires progressively deepening the knowledge mining process for finding refined knowledge from data. The necessity for mining multiple-level association rules or using taxonomy information at mining association rules has also been observed by other researchers [1].

Apriori algorithm is the best algorithm for single level association rules mining [7,8]. 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.

61

62 2. PROBLEM STATEMENT

63

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.

69

70 **Table 1. Transactional database**

71

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

72

73

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.

77

78

79

80

81

82

Table 2. Item master database

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

83

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. 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

159 **Fig. 1. Data.dat file**
160

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

163

164 **4. CLEANING OF DATA**

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

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

173 The algorithm of data cleaning is given in fig. 2. The algorithm of data cleaning takes the
174 data.dat file as input. It opens in read mode and creates and opens out.dat file in write
175 mode. It reads the data.dat file and checks for space and new line character. These
176 characters are the separators between two codes. It stores these codes and writes them into
177 out.dat if the count of the code is less than 6 digits.

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

181

182

```
183 Algorithm Data_Cleaning(data.dat)
184 {
185   Open data.dat file in read mode
186   Create and open out.dat in write mode
187   i=0
188   While( data.dat)
189   {
190     read data.dat into x
191     if x= ' ' or x= new line then
192       if i=1 then
193         for j =1 to 5 do
194           item[i]=item[i-1]
195       if i=2 then
196         for j =2 to 5 do
197           item[i]=item[i-2]
198         for j=0 to 5-i do
199           write into out.dat
200     else
201       write into out.dat
202     i=i+1
203   }
204 }
```

199

200 **Fig. 2. Algorithm of data cleaning**

201

202 **5. ALGORITHM**

203

204 Apriori algorithm is a classic algorithm for finding frequent item sets and single level
205 association rules [10]. A fast implementation of Apriori algorithm is presented using the trie
206 data structure in [9]. Bodon implementation generates frequent item sets and association
207 rules of single level. It does not generate the association rules of second level.

208 This Bodon implementation has been modified for finding the association rules of second
209 level. To facilitate the process of finding the level of association rules one argument has
210 been added. The necessary modifications are also done to process this new argument. One
211 additional function is added to separate the code of input file. After separating the coded
212 inputs, it calls the function to generate the association rules according to their required level.
213 This new addition of code is shown in fig. 3.

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

218

219 Loop till character is greater or equal to '0' and character is less than equal to '9'.

220 (i) split or extract the positional character of file pointer.

221 (ii) add the current character with previously produced value in step 1 (i).

222 (iii) get character of incremented index.

223 (iv) increment the position.

224 If level = 1 then

225 (a) converting int [item] to char [items] i.e. Casting item (int type) to items (char).

226 (b) extracting the product digits from item.

227 Else if level = 2 then

228 (a) converting int [item] to char [item] ie. Casting item (int) type to items (char).

229 (b) extr acting the brands digits (code) from item.

230 Else if level = 3 then

231 (a) converting int [item] to char [item] ie. Casting item (int) type to items (char).

232 (b) extr acting the brands digits (code) from item.

233

234 If produced or brand code is NULL or ZERO then

235 Item = 0

236 If position isnot NULL or ZERO then

237 Insert item into temporary basket.

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)

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 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

277 7. CONCLUSION AND FUTURE SCOPE

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

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

285

286

287 REFERENCES

- 288 1. Jiawei Han and Yongjian Fu, "Discovery of Multiple-Level Association Rules from Large
 289 Database", Proceeding of the 21st VLDB Conference Zurich, Swizerland, 1995, pp.420-431.
- 290 2. David W. Cheung, Vincent T. Ng and Benjamin W. Tam, "Maintenance of discovered
 291 knowledge: A case in Multi-Level Association Rules", proceeding of 2nd international
 292 conference on knowledge discovery and data mining in 1996. pp.1-10.
- 293 3. Rakesh Agrawal and John C. Shafer, "Parallel mining of association rules: Design,
 294 Implementation and Experience", IBM Research Report, IBM research division Almaden
 295 Research Center, San Jose, CA, 1996.
- 296 4. Piatetsky-Shapiro, G. (1991), Discovery, analysis, and presentation of strong rules, in G.
 297 Piatetsky-Shapiro & W. J. Frawley, eds, 'Knowledge Discovery in Databases', AAAI/MIT
 298 Press, Cambridge, MA.
- 299 5. R. Agrawal; T. Imielinski; A. Swami: Mining Association Rules Between Sets of Items in
 300 Large Databases", SIGMOD Conference 1993,pp. 207-216
- 301 6. Bing Liu,Wynne Hsu and Yiming Ma, "Mining association rules with multiple minimum
 302 supports", KDD-99 San Diego CA USA, pp.337-341.
- 303 7. R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules Between Sets of Items
 304 in Large Databases", Proceedings of the ACM SIGMOD Conference on Management of
 305 data, pp.207-216, May 1993.
- 306 8. R. Agrawal, and R. Srikant, "Fast Algorithms for Mining Association Rules", In Proc.
 307 VLDB 1994, pp.487-499.
- 308 9. F. Bodon, "Fast Apriori Implementation", Proceedings of the IEEE ICDM Workshop on
 309 Frequent Itemset Mining Implementations, 2003.
- 310 10. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of
 311 association rules. In Advances in Knowledge Discovery and Data Mining, 1996, pp. 307.328.