RESEARCH ARTICLE OPEN ACCESS

Reviewing execution performance of Association Rule Algorithm towards Data Mining

Dr. Sanjay Kumar[1], Abhishek Shrivastava[2]
Associate Professor[1], Research Scholar[2]
Jaipur National University
Rajasthan – India

ABSTRACT
Multilevel association rules use multi leveled approach to retrieve in depth insight into information extracted. Apriori algorithm explores the single level association rules. Here Fast Apriori implementation is enhanced to achieve new algorithm for generating multilevel association rules. In this study the performance of this new algorithm is analyzed in terms of execution speed in timed unit of fraction of seconds.

Keywords:- multiple-level association rule, fast Apriori implementation review, minimum support, minimum confidence, data coding, data cleaning, Time complexity, data mining, algorithm.

I. INTRODUCTION

Association rule approach for Data Mining is a popular and well researched method for extracting interesting relations between data values in large databases. With wide applications of computerized

Automated data collection tools such as ATMs, GPS etc, massive amounts of transactional data gets continuously collected in databases. The interesting association relationships discovered amongst massive data will help decision support systems. 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 apps 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. Coherent rules are discovered based on the properties of propositional logic, and therefore, requires no background knowledge to generate them. From the coherent rules discovered, association rules can be derived objectively and directly without knowing the level of minimum support threshold required[15].

In this study, the performance of our newly enhanced algorithm for finding frequent apps sets and mining different level of association rules has been analyzed in terms of execution speed measured in time for sample datasets. The different datasets available on Frequent Appset

Mining Dataset Repository has been used as a sample for the study.

II. USER WORKING ANALYSIS

Q R Codes are used for every app downloaded from internet. It facilitated the automatic reading of apps details using the QR code scanner in mobile / tablet or any smart device. QR code for an app is a design pattern visible on electronic display or can be displayed as printout. The transaction database of any app downloading website such as google play store records the device id and the set of QR codes downloaded against each device id. The sample transaction table is shown in table 1.

Table 1. Transactional database

Device id	{QR codes}
123	{121, 102, 876}
124	{121, 102, 121}
…	……………………………..

The app master table contains the details of apps against the each QRcode. If QRcode is considered as a sequence number then mapping of app details from apps master database to transaction database is required to produce some meaningful database. Table 2 shows the sample apps master database.

Table 2. Apps master database

QR code	Category	Details	Manufacturer	Version	Price ($.)
121	Games	Angry birds	ROVIO	Rio	76
102	Social	Facebook	Facebook	Apps	Free
………………………………………………………….…….	….	……..	……..	……….	

Apps master database contains the complete details of apps against each QR code. The QR code 121 represents the apps details as

QR code	Category	Details	Manufacturer	Version	Price ($.)
121	Games	Angry birds	ROVIO	Rio	76

This apps master is providing three level of concept hierarchy. First level the apps category, second level the app Manufacturer and the third level is version. By 3rd level association rules, the association between Games category Rio version with Social category Facebook Apps will be explored. The execution time of the modified algorithm for finding different level association rules from different databases using fast Apriori implementation is recorded and analyzed in terms of execution speed in seconds.

III. CODING OF DATA

This algorithm functions on database containing encoded datasets. In this study, the six digits code has been used for every app downloaded. The six digits of the code has been divided into three level hierarchy so two digits per level. As for sampling in this study, maximum hundred categories can be coded starting from 00 to 99. Every version of apps can have maximum of hundred manufacturer and every manacturer for given category can have maximum of hundred version options. This coding can be flexible in future studies. Using three tables of code and apps category, manufacturer and versions, the coding of the database has been done easily.

Sample coding scheme is shown in table 3. Every apps category is represented by two digit code.

By this approach, maximum of hundred apps category can be coded. So after reading the apps category the program which is responsible to generate the codes will generate two digits code for every apps category.

Table 3. Coding scheme for apps categories

S.No.	apps	Code
1	Games	10

2	Social	11
3	Antivirus	12
4	Books	13
5	Entertainment	14

Table 4 is showing the sample coding scheme for manufacturer of apps version of Angry Birds. So every Apps category can have hundred manufacturers. The program which is responsible to generate the code will put two digits code for the manufacturer name of apps category. For example for manufacturer ROVIO of apps category GAMES, the code is 20.

The table 5 is showing the sample code for versioning options of apps. Generally apps comes in various versioning options. By this coding scheme, maximum of hundred versioning option are available to code the versioning options for every manufacturer of apps category. For example free version of manufacturer ROVIO of apps category GAMES 102000.

Table 4. Coding scheme for brands of apps under category Games

S.No.	Apps	Code
1	ROVIO	20
2	FaceBook	21
3	MicroSoft	22
4	SmartApps	23
5	PandaApps	24

Table 5. Coding scheme for versions of manufacturer ROVIO apps category GAMES

S.No.	App details	Code
1	Angry Birds Basic	00
2	Angry Birds Rio	01
3	Angry Birds Space	02
4	Angry Birds Christmas	03

The complete coding scheme of apps is shown in table 6. The program will generate six digits code for every apps purchased. For example, 102001 is the code for apps category Games, apps manufacturer Rovio and versioning of Angry Birds Rio. 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 apps with manufacturers

S.No.	Item with manufacturer	Code
1	Rovio Games Angry Birds Basic	102000
2	Rovio Games Angry Birds Rio	102001
3	Rovio Games Angry Birds Space	102002
4	Rovio Games Angry Birds Christmas	102003

Table 7 is displaying the sample transaction table of the app store. Transaction id is assigned against each download from the store. For example the first customer downloaded the game of ROVIO manufacturer in FREE version, antivirus of AVG manufacturer in Standard version & Whatsapp Social in free version.

Table 7. Transaction table

Transid	Apps purchased
1	{Game(Angry Bird Basic(Free)),Social(Whatsapp(Free)), AntiVirus(AVG (standard))}
2	{Social(Whatsapp(Free)), Entertainment(Musicon(Premium)),News(TOI(free))}

3	{Game(Angry Bird Rio ($59)),Social(Whatsapp(Free)), Entertainment(Musicon(Premium))}
4	{Game(Subway Surfer(Free)), Social(Whatsapp(Free)),AntiVirus(AVG(Internet Security))}
5	{{Game(Subway Surfer(Free)), Social(Whatsapp(Free)), Entertainment(Musicon(Premium))}

The software used to encode the database will generate the data.dat file generating each row contains one row of transaction table. The row number will represent the transaction id and the contents of the row will represent the apps purchased against that transaction id.

The sample of data.dat file is shown in fig. 1.

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

Figure 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. Similarly input file for 2nd level association rules is created and used to find frequent item sets and produce 2nd level association rules output.

IV. DATA PREPARATION & PURIFICATION

DATA PREPARATION & PURIFICATION of data is required for the databases which are already available. Another program takes care of this. It takes the .dat file as input and ensures standard data format before starting the algorithm functionality. It fills the missing digits by replacing following the defined dataset values. After the data purification process, it generates the new .dat file which has all six digit codes.

The program reads every generated 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 and n as the number of required digits in the output file. It opens in read mode and generates 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 stores these codes and writes them into out.dat if the count of the code is less than n digits.

This algorithm returns out.dat file which have all n digits coded into it. It completes our data purifying process. It is a complete input file so our algorithm for finding frequent item sets and association rules will work properly.

V. 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 apps 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 additional arguments are suggested. Necessary modifications are also done to process this new argument. One additional functionis 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.

VI. RESULTS

The new modified algorithm for finding frequent apps sets and mining multilevel association rules has been tested on these datasets. The results are recorded for level 1, level 2 and level3 for different minimum support. Every reading is recorded after taking the average of three executions.

```
Algorithm Data_Purify(data.dat, n)
{
```

```
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 n do
apps[i]=apps[i-1]
if i=2 then
for j =2 to n do
apps[i]=apps[i-2]
for j=0 to n-I do
write into out.dat
else
write into out.dat
i=i+1
}
}
```

Observations.

Minimum support has been varied from 0.5 to 0.01.

Observations show that

1. Execution time is increasing with increased number of levels
2. Execution time is increasing with reducing minimum support.

Hardware / Software Configuration used :

System Configuration : Core 2 Duo (1.6 GHz), with 2 GB RAM.
Operating system : Red Hat Linux.

Clock function is used to calculate the execution time.

Modus Operandi :

Our enhanced algorithm is used for testing the second and third level association rules and fast Apriori implementation given by Bodon has been used as add on for finding the first level association rules.

The T10ID100K and T40I10D100K datasets were generated BY the IBM Almaden Quest research group. The Kosarak dataset was provided by Ferenc Bodon and contains (anonymized) click-stream data of a hungarian on-line news portal. The Retail dataset was donated by Tom
Brijs and contains the (anonymized) retail market basket data from an anonymous Belgian retail
store. Data cleaning has been done for finding second and third level association rules.

Table 8 shows the results for T10ID100K dataset.

Table 8: T10ID100K dataset

Min Support	**Execution Time (Seconds)**		
	Level 1	**level 2**	**level 3**
0.50	0.38	0.44	0.51
0.05	0.76	0.93	1.04
0.03	0.86	1.02	1.17
0.02	1.00	1.15	1.30
0.01	1.28	1.45	1.59

Table 9 shows the results for T40I10D100K dataset.

Table 9: T40I10D100K dataset

Min Support	**Execution Time (Seconds)**		
	Level 1	**level 2**	**level 3**
0.50	1.42	1.64	1.90
0.05	4.37	4.91	5.32
0.03	5.02	5.50	6.00
0.02	5.99	6.57	7.07
0.01	35.26	36.14	36.29

Table 10 shows the results for Kosarak dataset.

Table 10: Kosarak dataset

Min Support	**Execution Time (Seconds)**		
	Level 1	**level 2**	**level 3**
0.50	0.84	0.88	1.02
0.05	1.82	1.95	2.17
0.03	1.86	1.95	2.19
0.02	1.92	2.02	2.26
0.01	2.10	2.19	2.43

Table 11 shows the results for Retail dataset.

Table 11: Retail dataset

Min Support	**Execution Time (Seconds)**		
	Level 1	**level 2**	**level 3**
0.50	0.36	0.40	0.47
0.05	0.73	0.84	0.99

0.03 0.74 0.86 1.00
0.02 0.75 0.87 1.01
0.01 0.81 0.96 1.08

Results show that the new algorithm is effective as the execution time for different levels of Association rule is proportionate with single level. Association rules of second level and third level provides the more specific information about the databases. Data purification Time is not included in the recorded execution time.

VII. CONCLUSION AND FUTURE SCOPE

The different datasets has been used in this study. The data purification has been done; prior to algorithm

Implementation for second and third level association rules.

The new modified algorithm for finding frequent item sets and mining different level association rules is executed on different datasets and execution time is recorded for different level with changed minimum support. The results are acceptable as execution time required to find second and third level association rules are proportionate to time required for determining the first level association rules. The execution time can be further improved by improving the data structure for recording the item sets.

REFERENCES

[1]. R. Agrawal, T. Imielinski; A. Swami: Mining Association Rules Between Sets of Apps 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 FrequentAppset Mining Implementations, 2003. Computer Science & Engineering: An International Journal (CSEIJ), Vol. 3, No. 4, August 2013

[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

[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.

[15]. RAKESH DUGGIRALA, P.NARAYANA. "Mining Positive and Negative Association Rules Using CoherentApproach", International Journal of Computer Trends and Technology- volume4Issue1- 2013 Page 1