Journal of Basic and Applied Engineering Research
Print ISSN: 2350-0077; Online ISSN: 2350-0255; Volume 2, Number 13; April-June, 2015 pp. 1116-1119
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/jbaer.html

A Numeric Approach to Mine Frequent Patterns from Very Large Database

Kapil Sharma[1] and Sachin Mittal[2]

[1]Delhi Technological University
[2]Samsung Electronics Pvt. Ltd.
E-mail: [1]kapil@ieee.org, [2]sachin.mittal08@gmail.com

Abstract—*This paper presents a new algorithm for mining frequent patterns from very large database. Mining Frequent pattern is one of the major activities in Data Mining field to extract the useful information from large databases. Frequent patterns are used by the big corporate houses/organizations to know the interest and purchasing trend of their customers and accordingly they plan their sales or marketing strategies. There exist algorithms to mine the frequent patterns. This paper presents a new efficient and fast algorithm to mine frequent patterns. Proposed new algorithms ie Magic Number Algorithm is based on the converting the database items to Numeric equivalents and which then help to apply mathematical calculations easily for fast conclusions of frequent Items. The proposed algorithm took 5µ seconds as compared to 9345µ seconds taken by Apriori Algorithm with same experimental setup and environment conditions.*

1. INTRODUCTION

This algorithm deals with Data Mining from large databases.

Active work on mining useful knowledge and information from very large database started in 90's [1] [2]. Frequent patterns are set of data items which occur more than a given threshold value in given large database set. Mining frequent patterns is probably one of the most important concepts in data mining because this tells about the trend of occurrence of data items. For example in large grocery stores this can tell about liking of customer for one data item as compared to other. By mining frequent patterns business houses can predict what kind of items needs to be presented to customer to increase the sale volume. Similarly in other areas also like scientific research, Universities data frequent patterns can be used to mine various useful facts for future strategy design.

Very large databases have millions of records and it is not possible to read all that information and extract the useful information. When strategists from various domains sit to analyze the available information, he need to know the trend of customers and users to design his strategy to increase the volume of sales etc. Frequent Patterns are most useful tool to help strategists to design their strategy. Frequent patters gives direct insight about users preferences and purchase habits in a particular domain of business like grocery store, electronics mega stores, e-commerce etc.

2. LITERATURE

Some of existing algorithms for frequent pattern mining are:

3. APRIORI

Apriori Algorithm [3] is based on making larger and larger Item sets. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database.

4. FP GROWTH

FP growth approach [4] is based on data structure FP-Tree for storing compressed, crucial information about frequent patterns. This approach smartly avoids costly candidate generation of Ariori. This was further elaborated in journal "Data Mining and Knowledge Discovery" [5]. It executes in multi passes. In the first pass, the algorithm counts occurrence of items (attribute-value pairs) in the dataset, and stores them to 'header table'. In the second pass, it builds the FP-tree structure by inserting instances. Items in each instance have to be sorted by descending order of their frequency in the dataset, so that the tree can be processed quickly. Items in each instance that do not meet minimum coverage threshold are discarded. If many instances share most frequent items, FP-tree provides high compression close to tree root. Recursive processing of this compressed growth tree is performed to mine the final dataset.

5. SETM

Algorithm SETM [6] uses only simple database primitives, viz., sorting and merge-scan join for mining patterns and association rules. It shows aspects of data mining can be carried out by using general query languages such as SQL, rather than by developing specialized black box algorithms.

6. PROPOSED ALGORITHM

Proposed new algorithm is based on fact that numbers can be processed faster than strings. But often our most of data is in String form like if we talk about grocery store then name of objects will be name of vegetables, fruits, or food items. But new proposed algorithm first converts these names of objects into numbers as per certain rule (not random). Once all objects or data items are represented by those numbers (or magic numbers) then this algorithm is applied and patterns can be mined in faster manner than the existing algorithms.

Table 1 Definitions

TS_n	Transaction Sum: Sum of all magic numbers of items purchased in nth transaction
MG_n	Magic Group: A Set having n such items for which total purchase count in complete database records is more than minimum support level
MS	Sum of magic number of all items in Magic Group currently under consideration
N1	Total number of transactions
N2	Total number of Items in magic group

same. Let’s call these numeric values a magic numbers.

Calculate sum of magic numbers of all N transactions. eg. TS_1 , TS_2 etc

```
magic_number(N1,N2)
if (N2 < 2) break;
count = 0;
for all $TS_{n,}$ n = {1,2,3…N1}
If( ($TS_n$ – MS) is Zero or SUM of magic numbers)
count++;
If (count >= min_support)
    $MG_{N2}$ & all its subsets are Frequent Pattern
    break;
    else if (j == N1){ /* reaches final transaction */
    For N2 subsets of N2-1 members in MG
            N2 = N2-1
        magic_number (N1,N2);
end;
```

Answer: All subsets of MG_{N2}

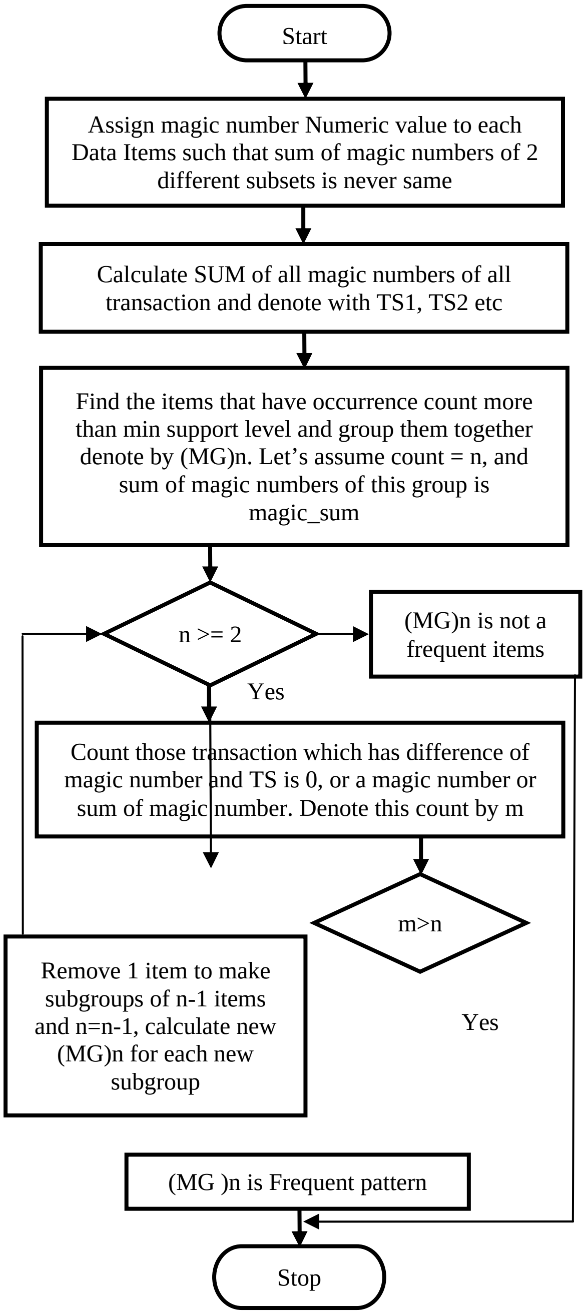

Start
Assign magic number Numeric value to each Data Items such that sum of magic numbers of 2 different subsets is never same
Calculate SUM of all magic numbers of all transaction and denote with TS1, TS2 etc
Find the items that have occurrence count more than min support level and group them together denote by (MG)n. Let’s assume count = n, and sum of magic numbers of this group is magic_sum
n >= 2
(MG)n is not a frequent items
Yes
Count those transaction which has difference of magic number and TS is 0, or a magic number or sum of magic number. Denote this count by m
m>n
Remove 1 item to make subgroups of n-1 items and n=n-1, calculate new (MG)n for each new subgroup
Yes
(MG)n is Frequent pattern
Stop

Fig. 1: Flow Diagram

Assign numeric value to each item such that sum of numeric values of 2 subsets only be equal if data items in both are Algorithm Illustration

- Let’s assume a retail store chain like Wal-Mart , Big Bazar, Home-Plus etc, where they have various products like daily need things , cosmetic products , Stationary, Vegetables & Fruits.
- Owner company of these stores target is to mine the frequent patterns to know user preference and increase sales in future
- Below are 5 Transactions made at stationary section in a Retail store.
- Let’s apply Magic Number Algorithm to mine frequent patterns on below transaction table. Min support level for FP= 3

Table 2: Original Transaction Table

T1	**Eraser**	**Notebook**	**Pencil**	**Stapler**
T2	Stapler	Pencil	Notebook	
T3	Eraser	Notebook	Sharpener	
T4	Notebook	Sharpener	Pencil	Stapler
T5	Pencil			

- After Applying Magic Number Algorithm, Table with occurrence count will become like as follow.

Table 3: Transformed Transaction Table

#	Eraser	Note book	Pencil	Sharpener	Stapler	MAGIC SUM
T1	1	2	4	0	16	23
T2	0	2	4	0	16	22
T3	1	2	0	8	0	11
T4	0	2	4	8	16	30
T5	0	0	4	0	0	4
	2	4	4	2	3	

- Each Item is represented by its assigned magic number.
- Magic Sum of each transaction is shown in last column
- Now we can mine frequent pattern using Magic Number algorithm as follows.
- Notebook, Pencil, Stapler has selling more than min Support 3. Sum of magic number of these is 2+4+16 = 22.
- There are 3 transaction that have Sum more than 22. T1 = 23, T2 = 22, T3=30
- T1: 23 – 22 = 1 , and 1 is also magic number, or sum of magic nos, thus T1 has <Notebook,Pencil,Stapler> together.
- T2: 22– 22 = 0 , T2 has exactly <Notebook,Pencil,Stapler> together.
- T3: 30-22 = 8, 8 is also magic number, thus T3 has <Notebook,Pencil,Stapler> together.
- Finally it is proved that <Notebook,Pencil,Stapler> is Frequent Pattern as it has min support count = 3

7. SIMULATION

Simulation of algorithm is done using Android application programming on Android KitKat version 4.4. Implementation can be verified on any Android Mobile Device running on Android Kitkat version 4.4

Table 4: Development Environment

OS	Android 4.4 (KitKat)
SDK	ADT Build: v21.1.0-569685
CPU Capacity	ARM, 1.9GHz Quad Core
RAM	3 GB
API Level	17

In Simulation program below was 10 transaction made.

Input Data	**Transaction Detail**
T1	< Pencil, Eraser,Colors,Cutter>
T2	< Pencil, Eraser,Colors,Cutter>
T3	< Pencil, Eraser,Colors,Cutter>
T4	< Pencil, Eraser,Colors,Cutter>
T5	< Pencil, Eraser,Colors,Cutter>
T6	< Sharpener,Scale,Colors>
T7	<Sharpener,Notebook,Inkpot>
T8	<Colors,Cutter,Notebook>
T9	<Cutter,Inkpot>
T10	< Pencil, Cutter>

Output of Simulation Program:

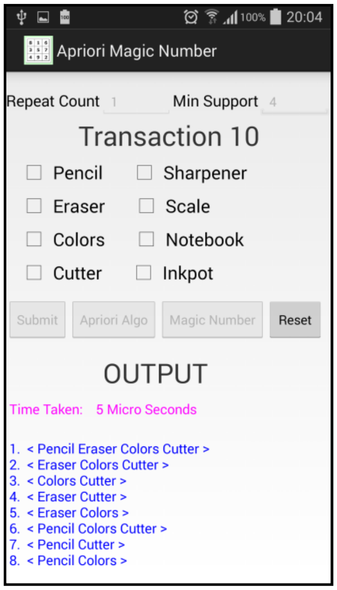
Output of simulation program is list of frequent patterns and time taken to compute the frequent patterns.

Magic Number algorithm is found superior to Apriori in terms of time taken to extract the Frequent patterns from same Input Data.

Time taken by Magic Number Algorithm- 5 µSeconds

Time taken by Apriori Algorithm- 934 µSeconds

Magic Number Algorith
Apriori Algorithm



Repeat Count 1 Min Support 4

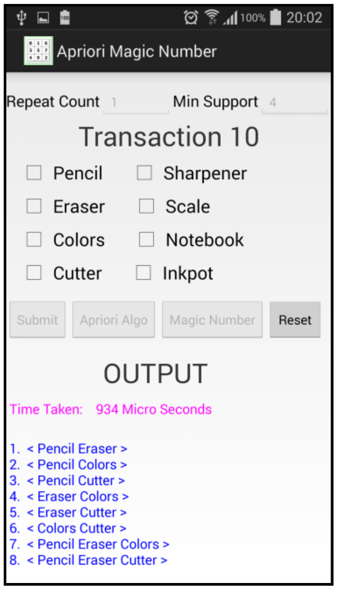
Transaction 10

□ Pencil □ Sharpener
□ Eraser □ Scale
□ Colors □ Notebook
□ Cutter □ Inkpot

OUTPUT

Time Taken: 5 Micro Seconds

1. < Pencil Eraser Colors Cutter >
2. < Eraser Colors Cutter >
3. < Colors Cutter >
4. < Eraser Cutter >
5. < Eraser Colors >
6. < Pencil Colors Cutter >
7. < Pencil Cutter >
8. < Pencil Colors >



Repeat Count 1 Min Support 4

Transaction 10

□ Pencil □ Sharpener
□ Eraser □ Scale
□ Colors □ Notebook
□ Cutter □ Inkpot

OUTPUT

Time Taken: 934 Micro Seconds

1. < Pencil Eraser >
2. < Pencil Colors >
3. < Pencil Cutter >
4. < Eraser Colors >
5. < Eraser Cutter >
6. < Colors Cutter >
7. < Pencil Eraser Colors >
8. < Pencil Eraser Cutter >

8. CONCLUSION

Proposed Magic Number algorithm is verified by programming simulation using Android Mobile Programming and is able to extract the correct information of frequent patterns. Also, performance of this algorithm is found superior to the existing Apriori Algorithm on same experimental setup and environmental conditions.

This algorithm can be used as a new method to discover frequent patterns from very large databases.

REFERENCES

[1] R. Agrawal, T. Imielinski and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," in *ACM SIGMOD International Conference on management of data*, Washington DC, 1993.

[2] R. Agrawal, C. Faloutsos and A. Swami, "Efficient similarity search in sequence databases," in *Fourth International Conference*, Chicago, October 1993.

[3] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," in *20th International Conference on Very Large Data Bases*, Santiago, Chile, September 1994.

[4] J. Han, J. Pei and Y. Yin, "Mining Frequent Patterns without Candidate Generation," in *ACM SIGMOD international conference on management of data*, Dallas, 2000.

[5] J. Han, J. Pei, Y. Yin and R. Mao, "Mining Frequent Patterns without Candidate Generation:A Frequent-Pattern Tree Approach," *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53-87, 2004.

[6] H. Maurice and A. Swami, "Set-Oriented Mining for Association Rules in Relational Databases," in Eleventh International Conference on Data Engineering, Taipei, 1995.