Modified Prime Number Based Partition Algorithm

Mr. Praveen Kumar Mudgal, Prof. Ms. Shweta Modi

Abstract— Frequent pattern mining is always an interesting research area in data mining to mine several hidden and previously unknown pattern. The better algorithms are always introduced and become the topic of interest. Association rule mining is an implication of the form X implies Y, where X is a set of antecedent items and Y is the consequent items. There are several techniques have been introduced in data mining to discover frequent item sets. This paper describes and takes an approach for mining frequent pattern and suggests some modification. The new algorithm uses both the concepts of top-down and bottom-up approach.To calculate the support count prime representation of item sets is used .It enables to save time in calculating frequent item sets. Through this Efficiency of system improves when the frequent item sets are generating in lesser time.

Index Terms—Association Rule Mning,Cluster Based Partition Algorithm,Data mining,Frequent patterns,KDD,Support Count,Prime numbers.

1 INTRODUCTION

The method of association rule mining was given by R. Agrawal in 1993.It can show the association of various products by generating frequent item sets. The data mining requires, need to store wide variety of data to be stored in memory for future processing .It takes several years to accomplish this large storage. Because the size of data is maintained and processed basically captures more then tera bytes of space in memory. This kind of large volume of data may originate from business enterprises and scientific research. Data mining is the process of extracting interesting and undetected patterns from large storage after their processing. Data mining is a one step of the process known as Knowledge Discovery in Database; through this process relative and previously unknown and needful information can be extracted. For extracting such kind of important information data filtered through KDD process of data mining. This data helps in decision support, market analysis, deciding market policies, weather forecasting, medical diagnosis and many other applications.

2 ASSOCIATION RULE MINING PROBLEM

2.1 Concept of Association Rule Mining

Problem is based on the market basket problem. One can state the problem as follows: Let A= {a,b,c,…………,y,z} be the set of m different literals. Transaction database T is a collection of transaction. Each transaction contains a set of items {a,b,………….,y,z} is a subset of T.

Generally an association rules mining algorithm contains the following steps:

• The set of candidate k-item sets is generated by 1-extensions of the large (k -1) - Item sets generated in the previous iteration.

• Supports for the candidate k-item sets are generated by a pass over the database.

•Item sets that do not have the minimum support are discarded and the remaining item sets are called large k-item sets.

This process is repeated until no larger items sets are found.There are some algorithms which are used for mining frequent item set-

1) Apriori algorithm.
2) Partition algorithm
3) Pincer search algorithm

And many more algorithms are described after these three.

3 PRIME NUMBER BASED ASSOCIATION RULE MINING ALGORITHM

3.1 Concept of Prime number

This algorithm uses the concept of prime numbers to represent the items in the transaction. Each item is assigned a unique prime number. Each transaction is collection of items, so transaction is represented by prime product of uniquely assigned prime number of each item. Since the product of prime number is unique and modulo division of prime multiple of transaction by prime multiple of each item set can check the presence of item set in the transaction. Each division generates either remainder-

- If remainder = 0, item set is present in the transaction.
- If remainder <> 0, item set is present in the transaction.

The advantage of using prime number is presence of each item set can be calculated very quickly. Prime numbers representation method ge-

- *Mr.Praveen Kumar Mudgal is currently pursuing masters degree program in Software System RGPV University,India, E-mail: pmudgal1@mail.com*
- *Ms.Sweta Modi is currently HOD of masters degree program Computer Science in SRIT college ,India, E-mail: modi_shweta84@yahoo.com*

nerates only one number for one transaction.Each number is very easily operatable and storable in memory.Each number requires less processing time to calculate support count because each item set is unique.

Table 1: Sample Database

Tid	Transaction
A	2,3,4,5
B	1,2
C	2,3,4,5
D	2,4,6,8
E	2,3,13
F	3,4,5,6,7
G	6,7,8,9,12
H	10,11,12,13
I	12,13
J	10,13
K	8,9,10
L	1,3,5,7,12

In this algorithm each item is assigned a unique prime number.That takes part in calculating support count of each item sets. As shown is table 2.

Table 2: Equivalent Prime Assignments

Items	Equivalent Prime
1	2
2	3
3	5
4	7
5	11
6	13
7	17
8	19
9	23
10	29
11	31
12	37
13	41

3.2 Prime multiple representations

Reason to adopt Prime numbers in our new approach is to perform fast calculation for calculating support count of each item sets. The support count is calculated using modulo division operation between transactions prime product and items prime product. If the If the remainder is zero, it indicates that the item is in the transaction. If the remainder is non-zero, it indicates that the item is not present in the transaction. The support count of item set {2, 4} can be found by performing the modulo division of each transaction's prime product by the product '21' of item 2's corresponding prime number '3' and item 4's corresponding prime number '7' as shown in table 3. Next table 3 shows the prime multiple of each transaction.

Now it is the right time to calculate support count of item set {2, 4}. The support count of item set is found to be '3'.If modulo division operation of each transaction with item set {2, 4}'s prime multiple 21 gives remainder zero then it is counted as one support.This process can also be applied for calculating support of each item set.The time required for calculating the support count of each item set is same, because it requires only one single division operation.

Table 3: Prime multiple of each transaction

Tid	Transaction	Transaction Multiplication
A	2,3,4,5	3*5*7*11=1155
B	1,2	2*3=6
C	2,3,4,5	3*5*7*11=1155
D	2,4,6,8	3*7*13*19=5187
E	2,3,13	3*5*41=615
F	3,4,5,6,7	5*7*11*13*17=85085
G	2,6,7,9	3*13*17*23=15249
H	1,4,6,7	2*7*13*17=3094
I	2,13	3*41=123
J	10,13	29*41=1189
K	8,9,10	19*23*29=12673
L	1,3,5,7	2*5*11*17=1870

Table 4: Calculating support count of item set {2, 4}

Tid	Modulo Division (kmod21)K=	Remainder	Item's Presence
A	1155	Zero	Yes
B	6	Non-Zero	No
C	1155	Zero	Yes
D	5187	Zero	Yes
E	615	Non-Zero	No
F	85085	Non-Zero	No
G	15249	Non-Zero	No
H	3094	Non-Zero	No
I	123	Non-Zero	No
J	1189	Non-Zero	No
K	12673	Non-Zero	No
L	1870	Non-Zero	No

For calculating the support count two possibilities are as follows:

1) The support count for item set is greater than or equal to the minimal support count.
2) The support count for item set is less than the minimal support count.

For each possibility the two solutions are given below correspondingly:

1) It is maximal frequent item set and procedure end.
2) The subset of length equal to n=p/2 is generated and their support is calculated.

This again leads to the following possibilities.

1) If the support count of the set of size N is greater than the minimal support count, all possible supersets of size N+N/2 are generated.

2) If the support count is less than the minimal support count, the subsets of length equal to n/2 is generated.

This process ends when maximal frequent item set is found.

3.3 Previous algorithm:

The algorithm is divided in to two parts the first part which is the main algorithm which invokes the sub algorithm. The main algorithm creates equal partitions of transaction database and assigns each partition to sub algorithm. It then takes each partition and generates frequent item set.

3.3.1 Steps **of Algorithm for Master:**

1. Find the infrequent item sets of length 1 and store them in IF 1

2. Remove the infrequent 1-items as denoted by IF1 in all transactions.

3. Assign separate Prime Number Pj to each unique item ITj for n-items.

4. Represent the item sets in Prime Number Representation form as follows:

(a) Replace each Transaction's item ITj by Corresponding Prime Number Pj.

(b) Represent each Transaction Tj of Size m by the multiple Mj of all the prime number representation Pj of the items in the transaction (P1 x P2 x P3 x…….xPm) and store them in shared memory.

5. Find the size Maxlength of maximal size transaction in Database and put it in shared memory.

```
Lopfinal=false
6. For each node j in the cluster
Divide the transactions equally based on the number of nodes and assign to j-th Node.
Connect to j-th node's server program to initiate process.
End loop
7. for each node j in the cluster
Wait until result comes from j-th node
Combine the result from j-th node into finalresultset.
End loop.
```

3.3.2 Steps of Sub Algorithm:

1. Wait until master initiates process.

2. Read Transactions Tj from shared memory.

3. Read Prime number multiple Mj of Transactions from shared memory.

4. Read the size Maxlength of Maximal size transaction from shared memory.

```
SG=empty Where SG is the subset group
FrequentItemset = empty
Start = 1
End = Maxlength
j = Maxlength
Exitflg = 0
5. Do While Exitflg = 0
For each transaction Tj with size >= j do
For each itemset S of Transaction Tj with size j
Do
If S is not in SG AND if IF1's items are not a subset of S then
Find the Support count of itemset S
Mj mod K and counting its presence using the remainder and store it in SUPPORT
If SUPPORT >= minsupport then Add S to FrequentItemset
End If
Add S to SG;
End If
End loop
End loop
Clear the SG
If FrequentItemset is not empty
Start = j
j = Round ((Start + End)/2)
If j = End then
Send all Item set AllFrequentItemset to master
Exitflg = 1
Exit the Do loop
End if
Find the infrequent items in infrequent j-size
Item sets and add them to IF1
Add FrequentItemset to AllFrequentItemset
Clear the FrequentItemset
Else
End= j
j = Round (Start + End) /2
If j= Start -1 then
Send all Item set AllFrequentItemset to master Exit the Do loop.
End if
End if
Lopfinal=true
End Do loop
```

3.4 Modified Prime Number Based Partition Algorithm:

The new modified algorithm has suggested two modifications in this paper which works very efficiently. In the first modification sub algorithm select the records of size =j from the transaction database, at the same time it select the item set of size=j to generate the support count of the each item set. After the observation it is founded that number of transactions are very less as compared to support count, so there is no need to generate the support count this time, so time can be reduced this time. Another modification is that since we can generate multiple partitions of the data base after the each partition, frequent item set can be removed from further processing.

3.4.1 Steps of Algorithm for Master:

1. Find the infrequent itemsets of length 1 and store them in IF 1

2. Remove the infrequent 1-items as denoted by IF1 in all transactions.

3. Assign separate Prime Number Pj to each unique item ITj for n-items.

4. Represent the itemsets in Prime Number Representation form as follows:
(a) Replace each Transaction's item ITj by Corresponding Prime Number Pj.

(b) Represent each Transaction Tj of Size m by the multiple Mj of all the prime number representation Pj of the items in the transaction (P1 x P2 x P3 x.......xPm) and store them in shared memory.
5. Find the size Maxlength of maximal size transaction in Database and put it in shared memory.
Lopfinal=false
6. For each node j in the cluster
Divide the transactions equally based on the number of nodes and assign to j-th Node.
Connect to j-th node's server program to initiate process.
End loop
7. for each node j in the cluster
Wait until result comes from j-th node
Combine the result from j-th node into finalresultset.
Lopfinal=true
End loop.

3.4.2 Steps of Sub Algorithm:

1. Wait until master initiates process.

2. Read Transactions Tj from shared memory.

3. Read Prime number multiple Mj of Transactions from shared memory.

4. Read the size Maxlength of Maximal size transaction from shared memory.

```
SG=empty Where SG is the subset group
FrequentItemset = empty
Start = 1
End = Maxlength
j = Maxlength
Exitflg = 0
5. Do While Exitflg = 0
For each transaction Tj with size >= j do
For each itemset S of Transaction Tj with size j
If lopfinal=true then //first modification
Remove all itemset S with size j which are in allfrequentitemset.
If (noofrec < localsupport) Then//second modification
noofrec = 0
GoTo label5
End If
Do
If S is not in SG AND if IF1's items are not a subset of S then
Find the Support count of itemset S
Mj mod K and counting its presence using the remainder and store it in SUPPORT
If SUPPORT >= minsupport then Add S to FrequentItemset
End If
Add S to SG;
End If
End loop
End loop
Clear the SG
Label 5:
If FrequentItemset is not empty
Start = j
j = Round ((Start + End) / 2)
If j = End then
Send all Itemset AllFrequentItemset to master
Exitflg = 1
Exit the Do loop
End if
Find the infrequent items in infrequent j-size
Itemsets and add them to IF1
Add FrequentItemset to AllFrequentItemset
Clear the FrequentItemset
Else
End= j
j = Round (Start + End) / 2
If j= Start -1 then
Send all Itemset AllFrequentItemset to master Exit the Do loop.
End if
End if
End Do loop
```

4 Time Comparison In Both Algorithm

For the comparative study of previous algorithm and modified algorithm, we have taken a database of 1000 transaction of 13 items. In this analytical process we considered 1000 transactions to generate the frequent pattern with the support count 25% .We have repeated the same process by increasing the transaction, after the experiment on both approach, we have designed a graph and summarized a result in the following table 5.

Table 5: Recorded time with different parameters

Transaction	Cluster Based Partition Algorithm	Modified Prime Number Based Algorithm
200	26	18
400	46	41
600	64	55
800	92	82
1000	100	87

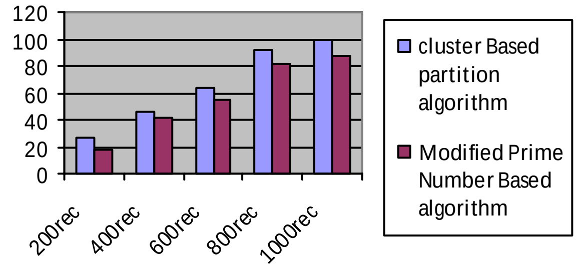

Fig 1. Performance evalution of Cluster Based Partition Algorithm and Modified Prime Number Based Algorithm.

5. Conclusion

In this paper, new algorithm for mining frequent item sets using prime number based partition approach was proposed. The concept of using prime number improves the performance of algorithm in terms of memory and time complexity. The pruning step in first scan reduces the size of transaction in memory besides the problem ofdiscovering association rules mining we have looked in to include the enhancement of data base capability.

Acknowledgment

We wish to thank Mr. Manish Tiwari for his insightful comments and suggestions.

References

[1] Arun K Pujari. Data Mining Techniques (Edition 5): Hyderabad, India: Universities Press (India) Private Limited, 2003.

[2] Margatet H. Dunham. Data Mining, Introductory and Advanced Topics: Upper Saddle River, New Jersey: Pearson Education Inc., 2003.

[3] Jiawei Han. Data Mining, concepts and Techniques: San Francisco, CA: Morgan Kaufmann Publishers.,2004.

[4] R.K. Gupta. Development of Algorithms for New Association Rule Mining System, Ph.D. Thesis, Submitted to ABV-Indian Institute of information Technology & Management, Gwalior, India, 2004.

[5] A. T. Bjorvand. Object Mining: A Practical Application of Data Mining for the Construction and Maintenance of Software Components. Proceedings of the Second European Symposium, PKDD-98, Nantes, France, 1998, pp :121-129.

[6] Akhilesh Tiwari, R. K. Gupta, D.P. Agrawal, Mining Frequent Itemsets Using Prime Number Based Approach. In Proc. 3rd International Conference on Advanced Computing and Communication Technologies (ICACCT), India, November 08-09,2008, pp: 138-141.

[7] M. Houtsma and A. Swami. Set Oriented Mining for Association Rules in Relational Databases. In Proceedings of 11th International conference on Data Engineering, 1995, pp 25-33.

[8] Agarwal R., Imielinski T., and Swami A. Mining associations between sets of items in massive databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington D.C. , May 1993, pp. 207-216.

[9] M. Houtsma and A. Swami, Set Oriented Mining for Association Rules in RelationalDatabases. In Proceedings of 11th IEEE International Conference on Data Engineering, 1995, pp : 25-33.

[10] Rakesh Agrawal and R. Srikant . Fast Algorithm for Mining Association Rules in Large Databases, Proceedings of the 20th International Conference on Very Large Databases, Santigo, Chile, 1994, pp 487-499.

[11] Akhilesh Tiwari, Rajendra K. Gupta, and Dev Prakash Agrawal Cluster Based Partition Approach for Mining Frequent Itemsets IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.6, June 2009.

[12]Dr.S.N.Sivanandam , Dr.S.Sumathi , Ms.T. Hamsapriya, Mr.K.Babu Parallel Buddy Prima – A Hybrid Parallel Frequent itemset mining algorithm for very large databases,Academic open internet journal.