



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

COMPARATIVE ANALYSIS OF APRIORI AND IMPROVED APRIORI ALGORITHM

Prafulla Bafna¹, Pravin Metkewar^{2*} and Angelina Gokhale^{3#}

¹Symbiosis Institute of Computer Studies and Research (SICSR),
Symbiosis International University (SIU), Atur Centre, Gokhale Cross Road,
Model Colony, Pune – 411 016, Maharashtra State, India

^{2*}Symbiosis Institute of Computer Studies and Research (SICSR),
Symbiosis International University (SIU), Atur Centre, Gokhale Cross Road,
Model Colony, Pune – 411 016, Maharashtra State, India

^{3#}Symbiosis Centre for Information Technology (SCIT),
Symbiosis International University (SIU), Rajiv Gandhi Infotech Park, Hinjewadi,
Pune – 411 057, Maharashtra State, India

Abstract: Mining data from a large database is one of the key research area. The Improved apriori algorithm helps in mining frequent items from the large database, by applying association rules between data in order to discover Knowledge from the large database. Due to the limitation and complexity of Apriori algorithm, a lot of research is being done to formulate new algorithms which will minimize the time and the number of database scans required for Knowledge Discovery through mining Association Rule among data from a large database. This paper compares and contrasts the behavior of Improved Apriori algorithm verses the existing Apriori algorithm. The Improved Apriori algorithm reduces the number of scans required for mining the frequent items from the large database and in turn mining the association rule between data. The uppermost transaction in the resultant matrix of the Improved Apriori algorithm contains the frequent item set found out by the existing Apriori algorithm. The Improved Apriori algorithm limits to consider a transaction in which all items are purchased. A Comparative analysis of both the algorithms results in an efficiently finding out the frequent item set.

Keywords: Apriori, Confidence, Support, matrix, AND operation, frequent Item Set, Probability

I. INTRODUCTION

The challenging task lies in extracting useful information from a large collection of data either from a data warehouse or from a database. Association rules are used for knowledge discovery and in managerial decision-making in the organization based on the results of associations among data thus stepping towards making a smarter system. In this regard, the first Apriori algorithm was proposed in the year 1994 by Agarwal and Srikanth to mine the frequent item set. It is an influential algorithm for mining frequent item sets using association rules.

Learning association rules means finding those items which are sold together more frequently than others. The Apriori algorithm generates frequent n-item set pattern but suffers from a large number of database scans required to identify those frequent items set thus making the task time consuming. The new improved apriori algorithm on the other hand is implemented with the main objective of minimizing the number of scans and the time taken to mine the frequent item set. The proposed efficient method dwells upon the concept of probability to evaluate the probability of candidate frequent item sets. Reduction in the number of database scans further decrements the runtime of the algorithm. A formula, in which association analysis is first used to generate parameters "a", "b" and then proceeds to get frequent patterns based on the best selected "a", "b" is proposed in this paper. A few other methods that confirm the obtained parameters "a", "b" are also mentioned. Implementation of this algorithm and the experimental analysis shows significant benefits of the proposed algorithm for different number of the item_sets and tuples in the database. Some factors like set size and set size frequency which in turn are being used to eliminate non significant candidate keys have been introduced. [2] With the use of these factors, the algorithm introduces a more efficient and effective way of minimizing candidate keys. These factors help in a rapid generation of possible association of frequent items. As far as the database iterations are concerned, the modified apriori provides less database scans compared with the original one that makes its execution faster. If the database is large, apriori algorithm takes too much time to scan the database. Improved apriori algorithm reduces the scanning time by cutting down unnecessary transaction records as well as reducing the redundant generation of sub-items while pruning the candidate item sets, [3] which can directly form the set of frequent item_sets and eliminate candidate having a subset that is not frequent. The Apriori algorithm is improved based on the properties of database. The typical apriori algorithm has a

restricted performance so it needs to be optimized with a variety of methods. The improved algorithm thus optimizes the typical apriori algorithm by reducing the size of the candidate set of k . [4] By pruning the candidate item sets, redundant generation of sub-items gets reduced, directly forming the set of frequent item sets and eliminating candidates having a subset that is not frequent. It also reduces the time duration for I/O and cuts down transaction records in the database. The performance of Apriori algorithm is optimized since information retrieval from massive data is faster and better. Although this improved algorithm has been optimized and become more efficient and it suffers from overhead to manage the new database. So it is necessary to have a different approach which will minimize the number of scans of the database. Another solution is the division of large database among processors. [5][6] Apriority algorithm is the classic algorithm of association rules, which enumerates all the frequent item sets. When this algorithm is applied, the large numbers of long patterns are declined dramatically. Finally, the improved algorithm is verified, the results show that the improved algorithm is reasonable and effective, thus extracting more valuable information.

Support: The support for an association rule $X \rightarrow Y$, is the percentage of transaction in the database which contains X and Y

Confidence: Confidence is the ratio of the number of transactions that contain X and Y to the number of transactions that contain X , where X and Y are two items from an item set

The above terms are used in the apriori algorithm to generate frequent item set in terms of association rules and both the terms are replaced by probability calculation in the improved apriori algorithm. Association rules are used in the last table of transaction to count concurrence of items. Eg. (Cosmetics and electronic equipments)[7]

Probability: Probability is a mathematical term that deals with calculating the likelihood of a given event's occurrence, which is expressed as a number between 1 and 0 [7]. In the improved apriori algorithm the probability measure is based on the occurrence of each data item from the total number of records or transactions.

II. RESEARCH METHODOLOGY

1. Apriori algorithm

With respect to an apriori algorithm an association rule can be termed as an event of items, so using probability will definitely come up with a fast algorithm (Because probability is measuring the possibility of occurrence of an event) to mine frequent item set. Matrix and Bitwise AND operations are used to minimize the time and number of scans. The improved apriori algorithm is better with respect to its performance. Thus it reduces the number of comparison required to mine the largest frequent items also once the largest frequent item set is found all its subsets will be identified and moved to frequent items set.

Improved Apriori algorithm

The existing Improved Apriori algorithm is based on the Bottom up approach following probability theory and Matrix manipulation in order to identify frequent item sets. Currently the objective behind this is to find out frequent item whose count would be arranged in descending order.

1.1 Steps for Apriori Algorithm.

Apriori algorithm is based on association rules. Purchasing one product along with the other represents an association rule. The database in which an association rule is to be found is viewed as a set of tuples, where each tuple contains a set of items.

Association rule generation is usually split up into two separate steps:

1.1.1 First, minimum support is applied to find all frequent item sets in a database.

1.1.2 Second, these frequent item sets and the minimum confidence constraint are used to form rules. By using the above steps it generates frequent item set pattern.

2. Improved apriori algorithm works in 2 phases that includes

2.1 Probability Matrix Generation

2.2 Bottom Up approach to mine frequent items set.

In the first phase :

2.1.1 In matrix $M1$: Rows in matrix represent transaction. Columns in matrix represent items. Each cell will have the values either 0 or 1 for representing presence of items in the transaction. (Entry value indicates the corresponding item is present or absent in transaction and value 0 indicates the corresponding item is not present in the transaction.)

2.1.2 For Probability matrix $M2$: 1 is replaced by the probability of occurrence of the corresponding

item present in the total number of transactions. Last two columns represent the probability matrix to hold total probability and count of elements in each row respectively.

2.1.3 The Probability Matrix will be rearranged as per the descending order of total probability. This leads to the generation of Sorted Probability Matrix M3.

In the second phase,

2.2 Non zero entries in the sorted probability matrix will be replaced by the value 1 leading to the generation of sorted probability Matrix M4. Frequent item set is found out from the entries in matrix M4.

Steps to Find out frequent item set

2.2.1. Select the first transaction from M4

2.2.2. Compare its total probability and count with the next transaction's total probability and count respectively.

2.2.3. If the current transaction's total probability is greater than the next transaction's probability then perform the BITWISE AND operation between the transactions.

2.2.4. If the resultant is equal to the first transaction structure then extract the item set of that transaction and all its subsets and move it to the frequent item set. Continue this process of Comparing and Bitwise AND operation with remaining transactions.

2.2.5. The same process will be repeated for remaining transactions until it finds an unseen transaction (unseen item) in the given data set.

2.2.6. If an unseen transaction is found out, then its item will be included in the frequent item set iff its BITWISE AND operation results into the same unseen transaction for all those transactions whose probability is higher.

2.2.7. For seen transactions there is no need to scan the database since it is a subset of frequent item set.

III. Illustration of Improved Apriori Algorithm

- Consider a Sample Dataset D , consisting of 30 transactions which represents the day wise sale of items.
- Scan the database and generate the initial Matrix M1, rows corresponding to Transactions and columns corresponding to Items respectively with cell values as either 0 or 1, indicating the presence of an item in the respective transaction.
- Generate the probability matrix M2 by replacing all non zero values of items with their corresponding probability of occurrence among the total number of transactions. Insert two more columns to represent the sum of probability and count of number of elements in each row.
- In Matrix M2, sort the table with respect to the occurrences column in descending order as shown in Matrix M3.
- Generate Matrix M4 by replacing each non-zero entry in M3 with 1 by maintaining the order of transactions similar to Matrix M3.

IV. Typical simulation of particular Transaction

Matrix M4 is the one which will be used to find out the frequent item set. The following operations are carried out on matrix M4.

- Select the first transaction D19 and extract D19 transaction Item set {Furniture, grains, electronic equipments, vegetables, cosmetics}
- Compare D19 transaction's total probability and count with next transaction's D1 total probability and count i.e **Probability (D19) $\geq$ Probability (D1) And count (D19) $\geq$ count (D1)** If the above condition satisfies,
- So by doing the Bitwise AND operation between D19 and D1. If the result of the Bitwise AND operation resembles the D19 Transaction, then consider Frequent Item Set which is present in D19, but condition fails for Bitwise AND then,
- Consider the next transaction as D23, extract item set from D23 {Furniture, grains, electronic, vegetables, cosmetics, clothes}, **Probability (D1) $\geq$ Probability (D23)** If the above condition satisfies, by performing the Bitwise AND operation between D1 and D23.
- If the result of the Bitwise AND operation resembles the D23 Transaction, then consider Frequent Item Set which is present in D23 { Furniture, grains, vegetables cosmetics, clothes} and conclude that largest candidate Item set {Furniture, grains, vegetables, cosmetics, clothes} now consider all its subsets as Frequent item Sets:
- Consider the next transaction as D9, extract item set from D9 {Furniture, grains, vegetables, cosmetics}, Check whether the item set is already present in Frequent Items set, if Item set already present in the Frequent Items Set, no need to further scan the database

- Select the Next Transaction D27 and the transaction Item set {Furniture, Electronic equipments, Vegetables, Clothes} which is not present in the Frequent Item set, so perform a bitwise AND with all those transactions having probability and count $\geq$ probability and count(D23) Bitwise AND with D19 resembles with D29, now updated frequent item set is { Furniture, electronic equipments, vegetables, cosmetics},
- Similarly bitwise AND of D6 does not resemble with any transaction, for D13 database scan is not required. Now again D26 is considered and its bitwise AND is done with all the transactions having probability and count $\geq$ probability and count(D28).
- So D23 transaction is the one in which Bit wise AND is done with D26 and results into D26, therefore, the updated frequent item set is {g, v, co, cl}. (Like-wise continue for D22-D27, D30-D27, D26-D5, D8-D28, D3-D28, D11-D26...) So as the list updates we come to a point where a frequent item set approaches to get its count (total number of items in the set)

In Table 2, comparative analysis of apriori and Improved apriori algorithm is given.

V. Limitations

As far as table 2 is concerned, Improved Apriori algorithm is better than Apriori algorithm but bitwise AND operation result will fail in case of Improved Apriori because AND output is 1 if all inputs are 1. This is the limitation of said algorithm.

VI. Future research

To strengthen the weakness of Improved Apriori algorithm in terms of large scale data, big tuple size, and calculation based on last matrix.

VII. Conclusion

Apriori algorithm applied on the above problem yields into frequent item set i.e., {Furniture, Grains, Vegetables, Cosmetics} which is the highest probability transaction consisting 4 items in matrix M4. The Improved Apriori algorithm on the other hand had to truncate a particular transaction which deals with purchase of all items because in that case each entry of that transaction for an item was represented by 1 i.e., (1,1,1,1,1) in matrix M4. On the basis of analysis so far we may conclude that bitwise AND operation result will fail in the above discussed case.

For the Apriori algorithm, as it has to find n-frequent item sets, total number of steps depends upon the total no of items but for the improved apriori algorithm total number of steps are predetermined.

As association rules have to be considered at each step to find out the frequent item set. Since the probability calculation had major operation and it is calculated only once and which is carried out as a last step. Also support and confidence is used only once in case of improved apriori but in case of apriori algorithm has to be considered at each step.

References

1. Li Pingxiang¹ Chen Jiangping² Bian Fuling, developed algorithm of apriori based on association analysis at State Key Laboratory of information Engineering in Surveying Mapping & Remote Sensing, Center of Spatial Information and digital Engineering, Wuhan University, 4300
2. Sheila A. Abaya, Association Rule Mining based on Apriori Algorithm in Minimizing Candidate Generation at International Journal of Scientific & Engineering Research Volume 3, Issue 7, July-2012 1 ISSN 2229-5518 IJSEER © 2012
3. Suraj P. Patil, U. M. Patil and Sonali Borse, The novel approach for improving apriori algorithm for mining association rule, ME Scholar, Department of Computer Engineering, RCPIT Shirpur, Dist Dhule (M.S.) India
4. Jaishree Singh*, Hari Ram, Dr. J.S. Sodhi, Improving Efficiency of Apriori Algorithm Using Transaction Reduction international Journal of Scientific and Research Publications, Volume 3, Issue 1, January 2013 1 ISSN 2250-3153
5. S.S.Phulari α , P.U.Bhalchandra α , Dr. S.D.Khamitkar α & S.N.Lokhande Understanding Rule Behavior through Apriori Algorithm over Social Network Data
6. Ashok Kumar D and Loraine Charlet Annie M.C.. Article: Web Log Mining using K-Apriori Algorithm. *International Journal of Computer Applications* 41(11):16-20, March 2012, Published by Foundation of Computer Science, New York, USA. BibTeX
7. M.Dunham, Data mining introductory and advanced topics, Pearson, 2008

Appendix:-

Table 1: Sample Data

Transaction Number(Day)	Items Purchased
D1	Furniture,Grains,Vegetable,Cosmetics,Clothes
D2	Equipments,vegetable,cosmetics
D3	Vegetables,Cosmetics
D4	Grains,Electronic Equipments
D5	Furniture,Grains,Electronic Equipments
D6	Furniture,Grains
D7	Vegetable,cosmetics
D8	Grains, Vegetable,Clothes
D9	Furniture, Grains,vegetable,Cosmetics,Clothes
D10	Electronic Equipments, Vegetables
D11	Furniture, Electronic Equipments
D12	Cosmetics,Clothes
D13	Electronic Equipments,Vegetables,Clothes
D14	Cosmetics
D15	Furniture,Electronic Equipments
D16	Furniture,Grains
D17	vegetable,Cosmetics
D18	Furniture,Clothes
D19	Furniture,Grains,Electronic Equipments,vegetable,Cosmetics
D20	Electronic Equipments,Clothes
D21	Furniture,Vegetable,Cosmetics
D22	Furniture,Electronic Equipments,Cosmetics
D23	Furniture, Grains,vegetable,Cosmetics,Clothes
D24	Furniture,Grains,Electronic Equipments
D25	Grains
D26	Grains,vegetable,Cosmetics,Clothes
D27	Furniture,Electronic Equipments,Vegetables,Cosmetics
D28	Electronic Equipments,Vegetables,Cosmetics

Table 2: Comparative Analysis

Algorithm	Total number of steps	Measure calculation based on	Usage Of probability to reduce the time	Consideration Of support and confidence
Improved Apriori	5(Always)	Last Matrix	Yes	Is replaced by probability value
Apriori	Incremented steps	All steps	No	At each step

Matrix M1: Sample Data converted into Binary

Day	Furniture	Grains	Electronic Equipments	Vegetables	Cosmetics	Clothes
1	1	1	0	1	1	1
2	1	0	0	0	1	1
3	0	0	0	1	1	0
4	0	1	1	0	0	0
5	1	1	1	0	0	0
6	1	1	0	0	1	1
7	0	0	0	1	1	0
8	0	1	0	1	0	1
9	1	1	0	1	1	0
10	0	0	1	1	0	0
11	1	0	1	0	0	0
12	0	0	0	0	1	1
13	1	0	1	1	0	1
14	0	0	0	0	1	0
15	1	0	1	0	0	0
16	1	1	0	0	0	0
17	0	0	0	1	1	0
18	1	0	0	0	0	1
19	1	1	1	1	1	0
20	0	0	1	0	0	1
21	1	0	0	1	1	0
22	1	0	1	0	1	0
23	1	1	0	1	1	1
24	1	1	1	0	0	0
25	0	1	0	0	0	0
26	0	1	0	1	1	1
27	1	0	1	1	1	0
28	0	0	1	1	1	0

Matrix M2: Probability computation and its count

Day	Furniture	Grains	Electronic Equipments	Vegetables	Cosmetics	Clothes	Occurrences (Total Probability)	Count
1	0.53	0.4	0	0.46	0.53	0.33	2.25	5
2	0.53	0	0	0	0.53	0.33	1.39	3
3	0	0	0	0.46	0.53	0	0.99	2
4	0	0.4	0.4	0	0	0	0.8	2
5	0.53	0.4	0.4	0	0	0	1.33	3
6	0.53	0.4	0	0	0.53	0.33	1.79	4
7	0	0	0	0.46	0.53	0	0.99	2
8	0	0.4	0	0.46	0	0.33	1.19	3
9	0.53	0.4	0	0.46	0.53	0	1.92	4
10	0	0	0.4	0.46	0	0	0.86	2
11	0.53	0	0.4	0	0	0	0.93	2
12	0	0	0	0	0.53	0.33	0.86	2
13	0.53	0	0.4	0.46	0	0.33	1.72	4
14	0	0	0	0	0.53	0	0.53	1
15	0.53	0	0.4	0	0	0	0.93	2
16	0.53	0.4	0	0	0	0	0.93	2
17	0	0	0	0.46	0.53	0	0.99	2
18	0.53	0	0	0	0	0.33	0.86	2
19	0.53	0.4	0.4	0.46	0.53	0	2.32	5
20	0	0	0.4	0	0	0.33	0.73	2
21	0.53	0	0	0.46	0.53	0	1.52	3
22	0.53	0	0.4	0	0.53	0	1.46	3
23	0.53	0.4	0	0.46	0.53	0.33	2.25	5
24	0.53	0.4	0.4	0	0	0	1.33	3
25	0	0.4	0	0	0	0	0.4	1
26	0	0.4	0	0.46	0.53	0.33	1.72	4
27	0.53	0	0.4	0.46	0.53	0	1.92	4
28	0	0	0.4	0.46	0.53	0	1.39	3

Matrix M3: Arranging Occurrences in Descending Order

Day	Furniture	Grains	Electronic Equipments	Vegetables	Cosmetics	Clothes	Occurrences	Count
19	0.53	0.4	0.4	0.46	0.53	0	2.32	5
1	0.53	0.4	0	0.46	0.53	0.33	2.25	5
23	0.53	0.4	0	0.46	0.53	0.33	2.25	5
9	0.53	0.4	0	0.46	0.53	0	1.92	4
27	0.53	0	0.4	0.46	0.53	0	1.92	4
6	0.53	0.4	0	0	0.53	0.33	1.79	4
13	0.53	0	0.4	0.46	0	0.33	1.72	4
26	0	0.4	0	0.46	0.53	0.33	1.72	4
21	0.53	0	0	0.46	0.53	0	1.52	3
22	0.53	0	0.4	0	0.53	0	1.46	3
2	0.53	0	0	0	0.53	0.33	1.39	3
28	0	0	0.4	0.46	0.53	0	1.39	3
5	0.53	0.4	0.4	0	0	0	1.33	3
24	0.53	0.4	0.4	0	0	0	1.33	3
8	0	0.4	0	0.46	0	0.33	1.19	3
3	0	0	0	0.46	0.53	0	0.99	2
7	0	0	0	0.46	0.53	0	0.99	2
17	0	0	0	0.46	0.53	0	0.99	2
11	0.53	0	0.4	0	0	0	0.93	2
15	0.53	0	0.4	0	0	0	0.93	2
16	0.53	0.4	0	0	0	0	0.93	2
10	0	0	0.4	0.46	0	0	0.86	2
12	0	0	0	0	0.53	0.33	0.86	2
18	0.53	0	0	0	0	0.33	0.86	2
4	0	0.4	0.4	0	0	0	0.8	2
20	0	0	0.4	0	0	0.33	0.73	2
14	0	0	0	0	0.53	0	0.53	1
25	0	0.4	0	0	0	0	0.4	1

Matrix M4: Converting Matrix M3 into Binary

Day	Furniture	Grains	Electronic Equipments	Vegetables	Cosmetics	Clothes	Occurrences	Count
19	1	1	1	1	1	0	2.32	5
1	1	1	0	1	1	1	2.25	5
23	1	1	0	1	1	1	2.25	5
9	1	1	0	1	1	0	1.92	4
27	1	0	1	1	1	0	1.92	4
6	1	1	0	0	1	1	1.79	4
13	1	0	1	1	0	1	1.72	4
26	0	1	0	1	1	1	1.72	4
21	1	0	0	1	1	0	1.52	3
22	1	0	1	0	1	0	1.46	3
2	1	0	0	0	1	1	1.39	3
28	0	0	1	1	1	0	1.39	3
5	1	1	1	0	0	0	1.33	3
24	1	1	1	0	0	0	1.33	3
8	0	1	0	1	0	1	1.19	3
3	0	0	0	1	1	0	0.99	2
7	0	0	0	1	1	0	0.99	2
17	0	0	0	1	1	0	0.99	2
11	1	0	1	0	0	0	0.93	2
15	1	0	1	0	0	0	0.93	2
1	1	1	0	0	0	0	0.93	2
10	0	0	1	1	0	0	0.86	2
12	0	0	0	0	1	1	0.86	2
18	1	0	0	0	0	1	0.86	2
4	0	1	1	0	0	0	0.8	2
20	0	0	1	0	0	1	0.73	2
14	0	0	0	0	1	0	0.53	1
25	0	1	0	0	0	0	0.4	1