# An Improved Apriori Algorithm Established On Probability Matrix

K. B. Agyapong, J. B. Hayfron-Acquah

**Abstract:** In this paper, the issue of scanning a large database is addressed by using probability matrix to generate the frequent itemset. The method eliminates the candidate having a subset which is not frequent. Currently in Computer Science and Data Mining, there are numerous mining algorithms of Associate Rule. Some of the supreme prevalent algorithms are the Apriori which extract frequent itemset from a large database. Although the Apriori algorithm is known to be the best for an Association Rule or Market Basket Analysis there are some challenges such as time wasting in scanning all the items found in the database through repetitive activities and the amount of memory space required as a result of having that large database being scanned. A comparison of the proposed algorithm with Apriori shows that the performance of the Improved Apriori is very promising.

**Index Terms**: Association Rule, Probability, Frequent Itemset, AND operation, Matrix.

## 1. INTRODUCTION

Data Mining is the indispensable progression of ascertaining hidden and thought-provoking pattern from enormous volume of documents from the data warehouse. By means of the development of technology of information, there is the requirement for digging out important information on large number of individuals from data set (Fayyad et al, 1996). Association rule is charity for data mining and to receipts important verdict grounded on administrative activities. The first algorithm proposed by Agrawal et al (1994) was used to mine the frequent itemset. Data Mining (also known as Knowledge Discovery in Database-KDD), according to Nasereddin (2009) includes the combination of practices from many chastisements such as database technology, machine learning and information retrieval, neural networks statistics etc. From the analysis of the abstracted pattern, decision making process can be done quite easily.

The improved algorithm see to solve the following two major problems associated with the Aprior Algorithm.

- To reduce the amount of comparisons in the direction of removing the largest regular item set for copy transaction
- To reduce the number of scanning and time needed in the direction of removing the regular item set when the most frequent item set is established.

## 2. METHODOLOGY

An improved Apriori algorithm is proposed to find frequent item set using probability and matrix. The proposed algorithm is in two (2) stages. The first stage is the preliminary matrix, which will be generated for the data set. In the matrix, rows depict transaction whereas columns show the items. The intersection of a column and a row is either T or F which will indicate occurrence of an item in the transaction or otherwise respectively. Two additional columns are added to store the total probability and the item count for each item. The second stage generates regular item sets directly from the probability matrix. The first transaction from the matrix is selected and its entire probability count is compared with the next transaction total probability and count respectively. If the next row count is more than or equal to the processing row count, then an AND operation is carried out among the rows, if the next row count is the same as the processing row item set construction, then the count value of support processing row item set is increased by one and carry on this action with AND operation through rest of the rows in the probability matrix. Next is to check the value of the total support. Pull out the item set and its subsets and shift them to frequent array list if it is greater or equal to the predefined min_support. Consider the 10 transactions and their item sets as shown in table 1.The first column shows the transaction identity whereas the second column shows the various items in the various transactions. The rows on the other hand depict the various transactions from A1 to A10.

***TABLE 1***
*SAMPLE DATASET*

| Transaction Identity | ITEM |
|---|---|
| A1 | 1, 2, 3 |
| A2 | 2, 3 , 5 |
| A3 | 4, 5 |
| A4 | 1, 2, 3 , 5 |
| A5 | 6 |
| A6 | 4, 5 |
| A7 | 1, 2, 6 |
| A8 | 5 |
| A9 | 1, 2, 3 , 5 |
| A10 | 2, 3 , 5 |


- *K. B. Agyapong is currently a Doctorate degree student in Computer Science. He holds an MPhil in Information Technology from Kwame Nkrumah University of Science and Technology, Kumasi, Ghana. Email: opanin007@yahoo.com*
- *J. B. Hayfron-Acquah is currently a Senior Lecturer in Computer Science at the Kwame Nkrumah University of Science and Technology, Kumasi, Ghana Email: Jbhayfron-acquah.cos@knust.edu.gh*

Scan the database to provide the initial matrix as seen in table 2. Every row corresponds to one transaction and column conforming to item respectively. Since transaction A1 has items 1, 2 and 3, these are indicated by 'T" in the first three columns to indicate the presence of the items. The remaining three columns had 'F' indicating the items did not occur in A1. The transaction A2 had 'F' in the first, fourth and the sixth columns indicating that items 1, 4 and 6 are not contained in transaction A2, the remaining had 'T' meaning there is a presence of item(s) in the transaction. The scan continues till the last transaction A10.The result is as shown in table 2.

***TABLE 2***
*MATRIX*

| Transaction | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| A1 | T | T | T | F | F | F |
| A2 | F | T | T | F | T | F |
| A3 | F | F | F | T | T | F |
| A4 | T | T | T | F | T | F |
| A5 | F | F | F | F | F | T |
| A6 | F | F | F | T | T | F |
| A7 | F | T | T | F | F | T |
| A8 | F | F | F | F | T | F |
| A9 | T | T | T | F | T | F |
| A10 | F | T | T | F | T | F |

From table 2, the 'T's and 'F's are then converted to probability values. Select the items in the various transactions and find its probability. The first transaction A1 is selected. The first item in column one is picked and divided by the total items, thus,1/6=0.16. The second item in the second column is also picked from the same transaction to determine the probability by dividing it by the total items, thus, 2/6=0.33. The next item in column three is also picked and its probability is determined as 3/6=0.5. The remaining columns do not have any item(s) as a result the probability is determined as 0/6=0, therefore the remaining columns have zero (0) each as their probability value. The total probability of transaction A1 is thus calculated as 0.16+0.33+0.50+0+0+0=0.99. The next transaction A2 is selected and the probability is determined as follows: The first column has no item recorded hence its probability is calculated as 0/6=0. The second column has an item, therefore its probability is determined as 2/6=0.33. The third column also has item thereby having its probability as 3/6=0.5. The fourth column has no item(s) hence its probability is given by 0/6=0. The fifth column has item(s) recorded hence its probability is determined as 5/6=0.83. The last column recorded no item(s) hence its probability is determined as 0/6 = 0. The total probability of transaction A2 is thus given as 0+0.33+0.5+0+0.83+0=1.66 This process is continued throughout the other transactions to determine the probabilities until the last transaction A10 is completed as shown in table 3.

***TABLE 3***
*PROBABILITY MATRIX*

| Transaction | 1 | 2 | 3 | 4 | 5 | 6 | Total Prob | Count |
|---|---|---|---|---|---|---|---|---|
| A1 | 0.16 | 0.33 | 0.5 | 0 | 0 | 0 | 0.99 | 3 |
| A2 | 0 | 0.33 | 0.5 | 0 | 0.83 | 0 | 1.66 | 3 |
| A3 | 0 | 0 | 0 | 0.66 | 0.83 | 0 | 1.49 | 2 |
| A4 | 0.16 | 0.33 | 0.5 | 0 | 0.83 | 0 | 1.82 | 4 |
| ~~A5~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~1.0~~ | ~~1.0~~ | ~~1~~ |
| A6 | 0 | 0 | 0 | 0.66 | 0.83 | 0 | 1.49 | 2 |
| A7 | 0 | 0.33 | 0.5 | 0 | 0 | 1.0 | 1.83 | 3 |
| ~~A8~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0.83~~ | ~~0~~ | ~~0.83~~ | ~~1~~ |
| A9 | 0 | 0.33 | 0.5 | 0 | 0.83 | 0 | 1.66 | 4 |
| A10 | 0 | 0.33 | 0.5 | 0 | 0.83 | 0 | 1.66 | 3 |

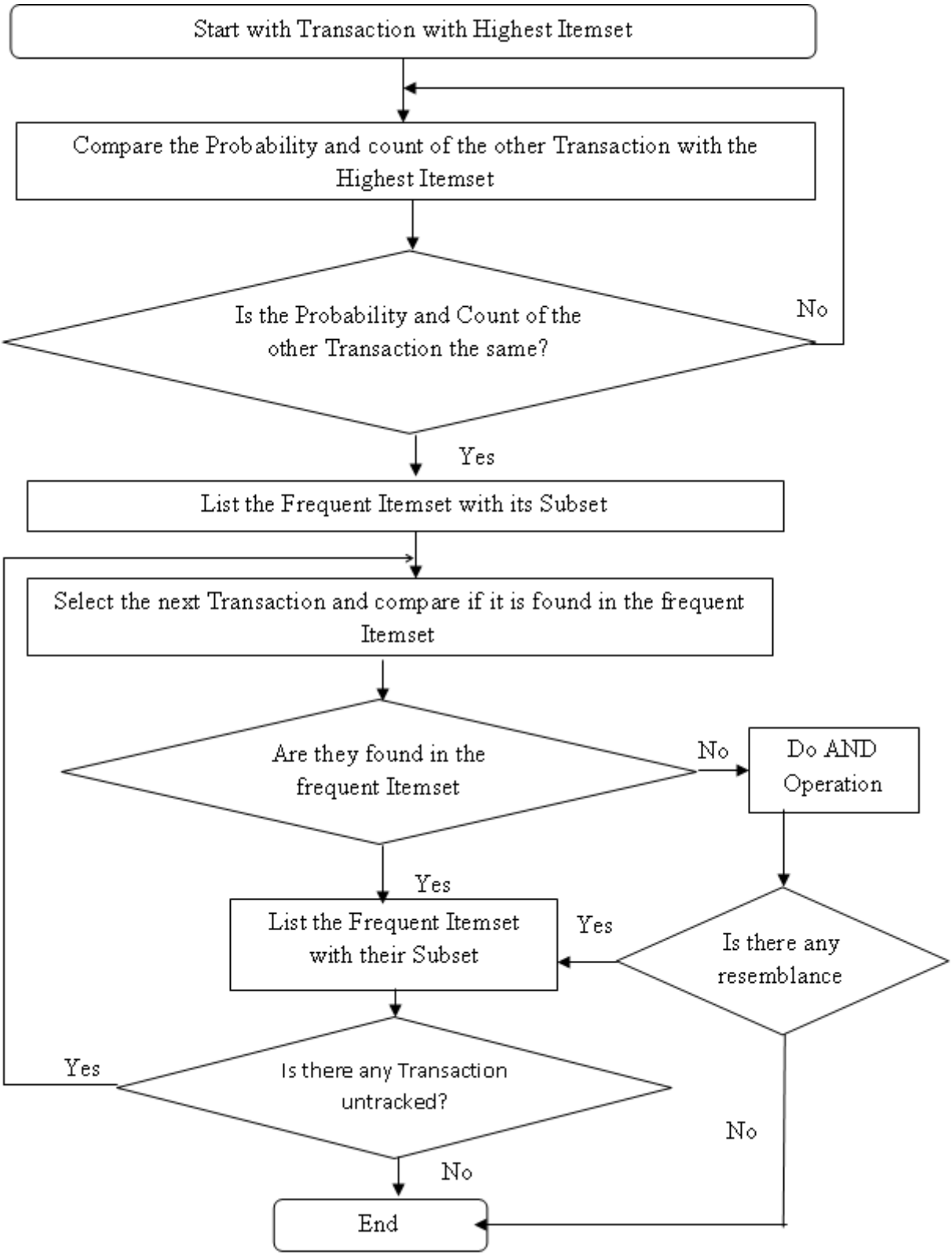
The Count are the number of non-zeros on each row. Since the minimum count is set to be 2, all transactions having a count of less than two (A5 and A8) are deleted. The result is as shown in table 4.

***TABLE 4***
*PROBABILITY MATRIX WITH COUNT GREATER THAN OR EQUAL TO TWO*

| Transaction | 1 | 2 | 3 | 4 | 5 | 6 | Total Prob | Count |
|---|---|---|---|---|---|---|---|---|
| A1 | 0.16 | 0.33 | 0.5 | 0 | 0 | 0 | 0.99 | 3 |
| A2 | 0 | 0.33 | 0.5 | 0 | 0.83 | 0 | 1.66 | 3 |
| A3 | 0 | 0 | 0 | 0.66 | 0.83 | 0 | 1.49 | 2 |
| A4 | 0.16 | 0.33 | 0.5 | 0 | 0.83 | 0 | 1.82 | 4 |
| A6 | 0 | 0 | 0 | 0.66 | 0.83 | 0 | 1.49 | 2 |
| A7 | 0 | 0.33 | 0.5 | 0 | 0 | 1.0 | 1.83 | 3 |
| A9 | 0 | 0.33 | 0.5 | 0 | 0.83 | 0 | 1.66 | 4 |
| A10 | 0 | 0.33 | 0.5 | 0 | 0.83 | 0 | 1.66 | 3 |

The processes in figure 1 are then applied to table 4.

***Figure 1:*** *Flowchart of the Algorithm.*



Following figure1, it is established that less time and less space were used to produce the final frequent item set as compared to the original Apriori Algorithm.

## 4. Experimental Experiments

Both algorithms were ran on an intel dual core PC apparatus with 2GB main memory and the program coded in java. The investigation standard dataset is from a material store database of 100,000 data records. The improved algorithm is compared with Apriori Algorithm using the same hardware resources, dataset and minimum support requirements. The output of the two algorithms is the same which demonstrates that our improved algorithm is proficient. The improved algorithm not only improves the algorithm of decreasing the size of the candidate itemset but also decreases the input/output outlay by cutting down transaction records in the database thereby not occupying a lot of memory space and less time was used from the results shown below.

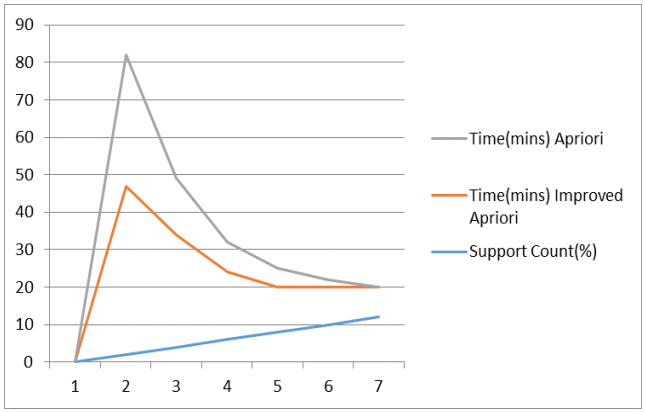



***Fig. 2:*** *Consuming time's comparison for different values of minimum support.*

## 5. Conclusion

In this paper an improved Apriori is proposed to decrease the time consumed in transactions scanning for candidate itemsets by reducing the number of transactions to be scanned. As seen in the surveillance, investigation and figure 2, the proposed algorithm performs much better than the prevailing Apriori. The improved algorithm can be used in many areas such as, elections, medicine, appearance dispensation, passport office, school database etc.

## 6. References

[1] Agrawal, R. and Srikant, R. (1994) . Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487-499, Santiago, Chile, September 1994.

[2] Gao Hongbin, Pan Gu, Huan Yiming. An enriched Apriori algorithm based on the character of the frequent itemsets[J]. Computer engineering and design.2007,28(10):2273-2275.

[3] H.H.O. Nasereddin,"Stream Data mining, "International Journal of Web Applications, vol. 1 no. 4 pp 183-190, 2009.

[4] M. Halkidi, "Quality assessment and uncertainty handling in data mining process" in Proc, EDBT Conference, Konstanz, Germany, 2000.

[5] Qiang Ma. Enriched Algorithm based on Apriori Algorithm[J]. Development and application of computer, 2010,23(2):6-7

[6] Rui Chang and Zhiyi Liu , " An Enriched Apriori Algorithm," ICEOE 2011, IEEE International Conference, vol. 1, pp v1-476 -v1-478.

[7] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to Knowledge discovery in databases" Al magazine, vol. 17 no. 3 pp. 37, 1996.

[8] Wanjun Yu; Xiaochun Waang; Erkan Wang; Bowen Chen;, "The research of enrichedapriori algorithm for mining association rules" Communication Technology , 2008.ICCT 2008 11th IEEE International Conference on, vol., pp. 513-516, 10-12 Nov . 2008.

[9] YuboJia, Guanghu Xia, Hongdan Fan, Qian Zhang and Xu Li, "An Enriched Apriori Algorithm Based on Association Analysis," ICNDC 2012, 3rd IEEE International Conference, pp208-211.

[10] Yiwu Xie ,Yutong Li Chunli Wang, MingyuLu. The Optimization and improvement of the Apriori Algorithm. In proc.Of 2008 Workshop on Education Technology and Training 2008 International Workshop on Geoscience and Remote Sensing.