Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.508 – 512

RESEARCH ARTICLE

Comparative Survey on Improved Versions of Apriori Algorithm

Sakshi Aggarwal[1], Dr. Ritu Sindhu[2]

[1]CSE Department, SGT Institute of Engineering & Technology, Gurgaon, India
[2]Associate Professor, CSE Department, SGT Institute of Engineering & Technology, Gurgaon, India
[1] sakshii.26@gmail.com

Abstract— In field of data mining, mining the frequent itemsets from huge amount of data stored in database is an important task. Frequent itemsets leads to formation of association rules. Various methods have been proposed and implemented to improve the efficiency of Apriori algorithm. This paper focuses on comparing the improvements proposed in classical Apriori Algorithm for frequent item set mining.

Keywords— Association rules, Apriori algorithm, Support, Confidence, Item set

I. INTRODUCTION

Association rule mining plays an important role in field of data mining because amount of data is increasing daily and mining the important and relevant information from this huge amount of data is a tedious task. Thus, mining the association rules helps in extracting the relevant information which thus helps the business people in decision making process. [1]

Association rule mining is useful in various areas like sales and marketing, storage planning, knowing the buying patterns of customer etc. Association rules are like if-then statements. Both if and then are the items in database. If is found in database while then is the combination of various items that acts as if items. If-then patterns are analysed to create association rules and support/confidence parameters are then used to identify strong association rules.

II. ASSOCIATION RULES

Association rules are defined as:
Let I= $\{I_1, I_2 ... I_N\}$
D=set of database transactions where T is the transaction and $T \subseteq I$
A=set of items
Then, an association rule is of the form A□B where $A \subset I$ and $B \subset I$ and $A \cap B = \Phi$

An association rule A→B holds with:
Support: number of transactions that contain both A and B i.e. $A \cup B$. It is represented by "s"

Confidence: it measures how items in B exists in transactions that contain A. It is represented by "c"

Support is given by probability P $(A \cup B)$ while confidence is given by probability P (B|A) [2] i.e.

$$\text{Support } (A \rightarrow B) = P(A \cup B)$$
$$\text{Confidence } (A \rightarrow B) = P(B|A)$$

Two step methods to find association rules are:
Step 1: find all frequent itemsets. An itemset is said to be frequent if item in each set satisfies minimum support value.
Step 2: generate strong association rules from frequent itemsets generated in above step.

Performance of association rule mining algorithm is dependent on step 1 explained above. Thus, counting the large itemsets is the focused area in association rule mining algorithms.

Example of Association Rules:
{Cell phone} → {Charger},
{Keyboard, CPU} → {Monitor, Motherboard},
So, an association rule is of the form X →Y, where X and Y is itemsets.
Example: {Cell phone} → {Charger}

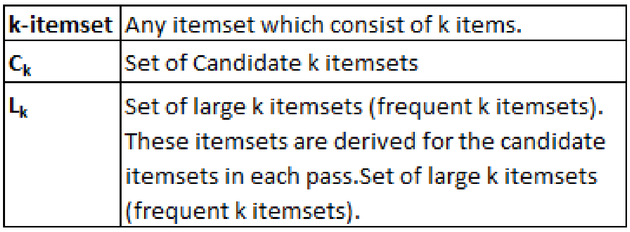
III. ASSOCIATION RULE MINING ALGORITHMS

All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified.

A. Classical Apriori Algorithm

Classical apriori algorithm is used to find frequent itemsets from the transactions in database. This algorithm says that an item (X) belonging to an itemset (I) is never large if itemset X itself is not large. It means that non-empty subset of a frequent item set must also be frequent. [1][3]

Important terms referred in classical apriori algorithm are:



k-itemset	Any itemset which consist of k items.
C_k	Set of Candidate k itemsets
L_k	Set of large k itemsets (frequent k itemsets). These itemsets are derived for the candidate itemsets in each pass.Set of large k itemsets (frequent k itemsets).

Fig. 1 Classical Apriori Algorithm

Basic steps of classical apriori algorithm are:

Generate and test: Candidate-1-itemset is found from scanning the transactions T of database D. Itemsets which satisfy the minimum support forms the frequent-1-itemset.

Join step: Self join of L_{k-1} is done with L_{k-1} i.e. $L_{k-1}*L_{k-1}$. From this self-join, candidate-k-itemset is found. Frequent-k-itemset is obtained from C_k which satisfies the minimum support count value.

Prune step: This step focuses on eliminating the some of the candidate-k-itemsets that are infrequent i.e. C_k is the superset of L_k but it may or may not be the case that all members of C_k are not frequent.

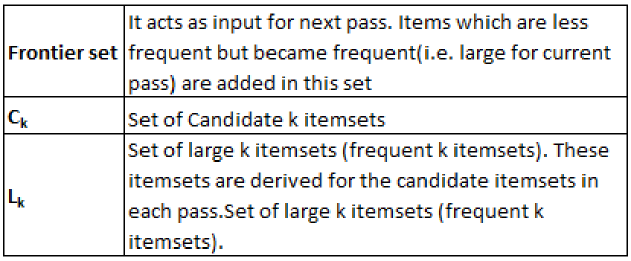
So, in pruning step, any infrequent itemset I of candidate-k-itemset will not be part of C_k. Join and prune step are repeated until no frequent itemsets can be generated.

Apriori algorithm successfully finds frequent itemset and strong association rules from database. But with the increase in database size, number of items increase which leads to below:
- Need more search space which thus increase the input/output cost
- Increase in computational cost to find candidate itemsets due to increasing scan on large database.

Therefore, many improvements have been proposed and implemented on Apriori algorithm which focuses on minimizing the limitations of increasing database size.

B. AIS Algorithm

AIS algorithm was given by [2]. While database is scanned for candidate itemset, C_k are generated and counted. Thus, to generate L_k, this algorithm makes multiple passes over C_k.



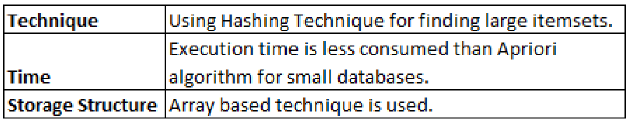
Frontier set	It acts as input for next pass. Items which are less frequent but became frequent(i.e. large for current pass) are added in this set
C_k	Set of Candidate k itemsets
L_k	Set of large k itemsets (frequent k itemsets). These itemsets are derived for the candidate itemsets in each pass.Set of large k itemsets (frequent k itemsets).

Fig. 2 AISAlgorithm

Frontier sets (F_k) are the key term in AIS algorithm. In each pass, frontier sets are extended and support is measured for each itemset. If support satisfies the minimum support threshold, itemset is added to C_k and also checked whether the itemset can be added to F_k. When no more items are added to F_k, algorithm terminates.

C. *DHP Algorithm*

DHP is Direct Hashing & Pruning Algorithm and it uses the data structure “Hash Bucket” for Ck generation. [4] Important points of DHP algorithm are:



Technique	Using Hashing Technique for finding large itemsets.
Time	Execution time is less consumed than Apriori algorithm for small databases.
Storage Structure	Array based technique is used.

Fig. 3 DHP Algorithm

Steps in DHP algorithm are:

Step1: generate frequent-1-itemset, generate has table for candidate-2-itemset.

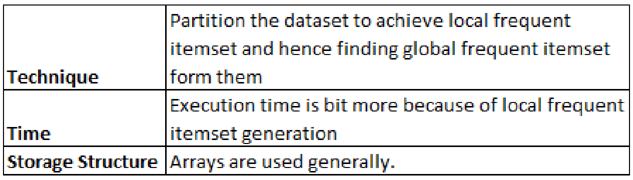
Step 2: find minimum transaction support (Min S_T) and add those items in C_k if k-item is hashed in hash table and whose value is greater than or equal to Min S_T.

Step 3: This step is same as step 2 with the change that hash tables are not used to add an itemset to C_k.

D. *Partition Algorithm*

Partition algorithm [5] focuses on reducing the passes on database. In apriori algorithm and DHP algorithm, each Ck generation needs multiple passes on database. But in partition algorithm, only two passes are needed as database is logically partitioned into N partitions and entire database is read at most twice only.

Important points of DHP algorithm are:



Technique	Partition the dataset to achieve local frequent itemset and hence finding global frequent itemset form them
Time	Execution time is bit more because of local frequent itemset generation
Storage Structure	Arrays are used generally.

Fig. 3 Partition Algorithm

IV. IMPROVEMENTS IN APRIORI ALGORITHM

Out of the various association rule mining algorithms discussed above, Apriori algorithm is the mostly used algorithm to find association rules from transactions in a database. But classical apriori algorithm suffers from various drawbacks. Few of them are as below;

- High I/O cost due to large size of candidate sets
- Large numbers of scans on database to find support of each item in each pass

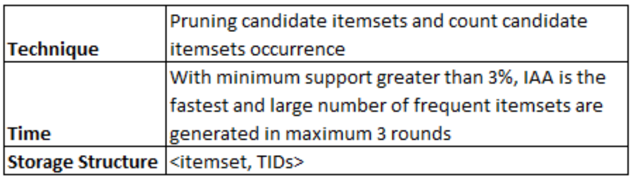
To improve the efficiency of Apriori algorithm, below are the points on which work is done to improve its efficiency:

- Decrease number of transactions in database
- Number of scans on database should be reduced
- Control high I/O cost

Studies on improved apriori algorithms are as below:

A. Improved Apriori Algorithm

IAA used count based candidate pruning method. In this method [6], theorem "if an item set is frequent, then every subset of the itemset will also be frequent" is given and deduction "k-dimensional L_k can be generated from L_{k-1} which differs only by one item"



Technique	Pruning candidate itemsets and count candidate itemsets occurrence
Time	With minimum support greater than 3%, IAA is the fastest and large number of frequent itemsets are generated in maximum 3 rounds
Storage Structure	<itemset, TIDs>

Fig. 3 IAA Algorithm

Steps of IAA are:

Step1: according to the type of data items, separate the data. Count data items when database is scanned

Step 2: Itemsets obtained in step 1 are pruned.

B. Association Rule Mining using Ant Colony Optimization

Ant Colony Optimization (ACO) [7] contains below two rules:

Rule 1: Local pheromone update rule used to construct the solution of the problem

Rule 2: Global pheromone update rule used in ant construction.

ACO algorithm focuses on minimizing the number of association rules. Apriori algorithm uses transaction data set and uses a user interested support and confidence value then produces the association rule set. These association rule set is discrete and continues. Hence weak rule set are required to prune.

C. Association Rule Mining using Genetic Algorithm

Genetic algorithm focuses on optimizing the rules. [8]This algorithm talks about condition attributes and decision attributes. GA is used to search the cut-points of continuous attributes.

This algorithm says that users are interested in associations from condition attributes and decision attributes. Thus, only the association rules with decision attributes are mined and thus numbers of association rules are optimized.

Steps to mine association rules are:

Step1: apply apriori algorithm to generate a rule set

Step 2: Separate strong rule set and poor rule set

Step 3: apply genetic algorithm to filter the rules which satisfies the minimum confidence value

Step 4: combine rules obtained in step 2 and step 3

D. Improved Pruning in Apriori Algorithm

Improved Pruning (IP) uses the concept of average support (S_{avg}) instead of minimum support (S_{min}) [9], Average support criteria generates probabilistic itemset rather than frequent itemset.

This algorithm explained that during pruning in classical Apriori algorithm, some important rules are pruned because user defined minimum support threshold filters the candidate itemsets as user does not know the support value at which association rules can be generated.

Thus, improved pruning method improves above by using the average support value rather than minimum support threshold.

Steps of improved pruning method are:

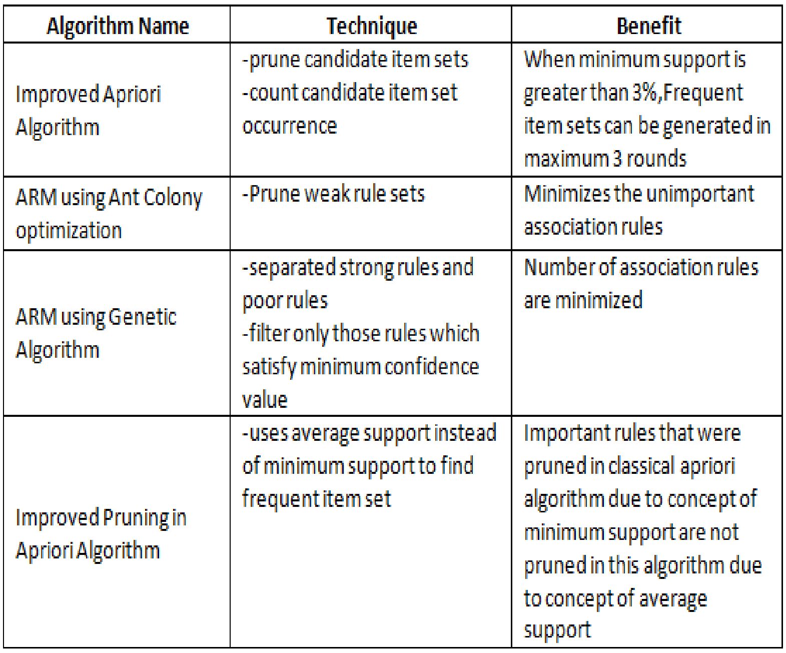
Step 1: calculate average support value

Step 2: insert itemsets having support value greater than S_{avg} into probabilistic itemset

Step 3: step 2 is repeated for all probability itemsets and thus, from these itemsets association rules are generated.

E. *Comparative Analysis*

A comparative view of above approaches on basis of improvements in technique/benefits is given as below:



Algorithm Name	Technique	Benefit
Improved Apriori Algorithm	-prune candidate item sets -count candidate item set occurrence	When minimum support is greater than 3%,Frequent item sets can be generated in maximum 3 rounds
ARM using Ant Colony optimization	-Prune weak rule sets	Minimizes the unimportant association rules
ARM using Genetic Algorithm	-separated strong rules and poor rules -filter only those rules which satisfy minimum confidence value	Number of association rules are minimized
Improved Pruning in Apriori Algorithm	-uses average support instead of minimum support to find frequent item set	Important rules that were pruned in classical apriori algorithm due to concept of minimum support are not pruned in this algorithm due to concept of average support

Fig. 4 Comparison of Algorithms

V. Conclusions

A survey on most recent work done in field of association rule mining to improve efficiency of Apriori algorithm is presented in this paper. On the basis of our study, we find that there are still issues that are needed to be studied to generate association rules with more efficiency. Some issues identified during our study in association rule mining are suggested below:

- Algorithms with one scan mining should be developed
- New techniques and applications for association rule mining should be identified and developed
- Techniques that are independent of database measurements should be identified.

Acknowledgement

I am thankful to my guide Dr. Ritu Sindhu (Associate Professor, SGT Institute of Engineering & Technology, MDU Rohtak) for taking time to read carefully drafts of this paper and provide the valuable comments. I am also grateful to all the faculty members who has motivated and supported me for preparing this manuscript.

References

[1] R. Agrawal, T. Imielinski, A. Swami, Mining Association Rules between Sets of Items in Large Databases, Proceedings of the 1993 *ACM SIGMOD Conference Washington DC*, USA, May 1993.

[2] R. Agrawal, T. Imielinski, A. Swami, Mining Association Rules between Sets of Items in Large Databases Proceedings of the 1993 *ACM SIGMOD Conference Washington DC*, USA, May 1993.

[3] J. Han, M. Kamber. "Data Mining: Concepts and Techniques", *Morgan Kaufmann Publishers, Champaign: CS497JH*, fall 2001.

[4] J. D. Holt and S. M. Chung, Efficient Mining of Association Rules in Text Databases, *ACM* 1999.

[5] A. Choubey, R. Patel, J.L.Rana, "A Survey of Efficient Algorithms and New Approach for Fast Discovery of Freqent itemset for Association Rule Mining", *IJSCE ,ISSN: 2231-2307*, vol. 1, issue 2,May 2011.

[6] W. Yong-qing, Y. Ren-hua, L. Pei-yu, "An Improved Apriori Algorithm for Association Rules of Mining" *IEEE*(2009)

[7] B. Patel, V. K. Chaudahri, R. K. Karan,YK Rana, "Optimization of association rule mining apriori algorithm using Ant Colony optimization" *International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307*, Volume-1, Issue-1, March 2011.

[8] R. Haldulakar, J. Agrawal," Optimization of Association Rule Mining through Genetic Algorithm", *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 3, Issue. 3, Mar 2011

[9] H. W., Zhigang L., L. Pan, R. S. XU and W. jiang, "An improved Apriori based algorithm for association rule mining" *IEEE Sixth international conference on fuzzy systems and knowledge discovery*, 2009, pp 51-55