JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-8, Aug 2018 E-ISSN: 2347-2693

Predicting Heart-Diseases from Medical Dataset Through Frequent Itemsets Using Improved Algorithm

V. Vijayalakshmi

Department of Computer Science, Government Arts College, Ariyalur, T.N, India

Corresponding author: v.viji08@yahoo.co.in

Available online at: www.ijcseonline.org

Accepted: 14/Aug/2018, Published: 31/Aug/2018

Abstract- In health concern business, data mining plays a significant task for predicting diseases. Mining association rule is one of the interesting topics in data mining which is used to generate frequent itemsets. It was first proposed for market basket analysis. Apriori algorithm is a classical algorithm of association rule mining and widely used for generating frequent item sets. This classical algorithm is inefficient due to so many scans of database. When the database is large, it will take too much time to scan the database and may produce a larger number of candidate item sets. To overcome these limitations, researchers have made a lot of improvements to the Apriori. In this paper, the authors proposed a method to predict the heart disease through frequent itemsets. Frequent itemsets are generated based on the chosen symptoms and minimum support value. The extracted frequent itemsets help the medical practitioner to make diagnostic decisions. The aim of our proposed technique is to obtain the frequent symptoms and evaluate the performance of new technique and compare with the existing classical Apriori with support count.

Keywords- Apriori, Frequent Diseases, Medical Data, Fuzzy Set, Fuzzy Intersection

I. INTRODUCTION

Data Mining is a recently emerging field, connecting the three worlds of Databases, Artificial Intelligence and Statistics. The computer age has enabled people to gather large volumes of data. Every large organization amasses data on its clients or members, and these databases tend to be enormous. The usefulness of this data is negligible if “meaningful information” or “knowledge” cannot be extracted from it. Data Mining answers this need.

Discovering association rules from large databases have been actively pursued since it was first presented in 1993, which is a data mining task that discovers associations among items in transaction databases such as the sales data [1]. Such kind of associations could be "if a set of items A occurs in a sale transaction, then another set of items B will likely also occur in the same transaction". One of the best studied models for data mining is that of association rules. This model assumes that the basic object of our interest is an item, and that data appear in the form of sets of items called transactions. Association rules are “implications” that relate the presence of items in transactions. The classical example is the rules extracted from the content of market baskets. Items are things we can buy in a market, and transactions are market baskets containing several items. Association rules relate the presence of items in the same basket, for example, “every basket that contains bread contains butter”, usually noted bread ⇒ butter. The basic format of an association rule is: An association is an implication of expression of the form A ⇒B, where A and B is disjoint itemset, i.e., A □□B=□.. The strength of an association rule can be measured in terms of its support and confidence. Support determines how often a rule is applicable to a given data set, while confidence determines how frequently items in B appear in transactions that contain A.

The formal definitions of these metrics are

Support s (A ⇒B) = □ (A □B)/N

Confidence c (A ⇒B) = □ (A □B))/□□ (A)

In general, association rule mining can be viewed as a two step processes:

1. Find all frequent itemsets: By definition, each of these itemsets will occur at least as frequently as a predetermined minimum support count.

2. Generate strong association rules from the frequent itemsets: By definition, these rules must satisfy minimum support and minimum confidence.

As the second step is rather straightforward and as the first step dominates the processing time, we explicitly focus the paper on the first step: the discovery of frequent item sets.

The remaining part of this paper is organized as follows: Section II discusses about Apriori Algorithm. Section III contains the basic concepts of Rough set theory. Section IV discusses Fuzzy Association Rules and Fuzzy measures. Section V elaborates the proposed method. Section VI contains worked example. Section VII discusses about the performance analysis of proposed algorithm compared with Apriori algorithm. Section VIII contains implementation. Section IX discusses conclusion.

II. RELATED WORKS

Agrawal proposed an algorithm, called AIS algorithm [1], for generating frequent itemsets. In the AIS algorithm, frequent itemsets are generated through iterations on scanning the database. The iteration terminates when no new frequent item-set is derived. After reading a transaction in the k^{th} iteration, the AIS algorithm computes the candidate k − itemsets by first deriving a set of (k−1) −itemsets which contains itemsets that are both in the frequent (k−1) −itemsets and in the transaction. However, Apriori algorithm has the limitation of producing a large number of candidate itemsets and scanning the database too many times [1].

Ayres.J introduced an effective pruning mechanism called depth first strategy to mine the sequential pattern in large database. This strategy defines the database in vertical bitmap format with effective support counting. For each item in the dataset, a vertical bitmap is constructed by which each data set transaction is represented as a bit. The value for items is set based on the item present in the transaction. The efficient support counting and candidate generation is obtained by partitioning the bitmap [2].

Changsheng Zhang and Jing Ruan have worked on the improvement of Apriori algorithm by applying dataset reduction method and by reducing the I/O spending. Changsheng and Jing Ruan have applied the modified algorithm for instituting cross selling strategies of the retail industry and to improve the sales performance [3].

Chen.J and Xiao.K presented bitmap itemset support counting (BISC) method [4].

Chen Chu-xiang et al. proposed R_Apriori algorithm which solves the problems of Apriori algorithm to improve the efficiency of the algorithm and is in promotion on certain significance [5].

Dongme Sun and Sheohue Teng has presented a new technique based on forward and reverse scan of database. It produces the frequent itemsets more efficiently if applied with certain satisfying conditions [6].

Feldman et al. presented a new method for computing occurrence frequencies of the various keywords labeling the documents [7].

Guan, J.W, Bell, D.A, Liu, D.Y proposed rough set approach to discovering knowledge is much simpler than the maximal association method [8].

Hanbing Liu presented a new association rule algorithm called ABBM (Association Rule Mining Based on Boolean Matrix algorithm) which transforms a transaction database into a Boolean matrix. It scanned the transaction database once, it does not produce candidate itemsets, and it adopted the Boolean vector “relational calculus” to discover frequent itemsets. In addition, it stores all transaction data in bits, so it needs less memory space and can be applied to mining large databases [9].

Jaisree Singh developed a Transaction Reduction Algorithm which reduced the scanning time by cutting down unnecessary transaction records as well as reduce the redundant generation of sub-items during pruning the candidate itemsets, which can form directly the set of frequent itemsets and eliminate candidate having a subset that is not frequent. But it has overhead to manage the new database after every generation of L_k. So, there should be some approach which has very less number of scans of database [10].

Kavitha.K proposed an efficient transaction reduction technique named TR-BC to mine the frequent pattern based on bitmap and class labels. The proposed approach reduces the rule generation by counting the item support and class support instead of only item support. Moreover, the database storage is compressed by using bitmap that significantly reduces the number of database scan. The rules are reduced by horizontal and vertical transaction and then finally combined rules are generated by eliminating the redundancy [11].

Logeswari.T et al. presented enhanced Apriori Algorithm which takes less scanning time. It is achieved by eliminating the redundant generation of sub-items during pruning the candidate item sets [12].

Pethalakshmi.A and V.Vijayalakshmi proposed an efficient count based transaction reduction approach for mining frequent patterns [13].

Ramaraj.E proposed a novel frequency itemsets generation algorithm called TRApriori that maintained its performance even at relative low supports. The advantages of TRApriori include interactive mining with different supports; faster execution time and infrequently used item are not stored and hence improves the size of the query data [14].

Sixue Bai and Xinxi Dai have presented a method called P-matrix algorithm to generate the frequent itemsets. It is found that the P-Matrix algorithm is more efficient and fast algorithm than Apriori algorithm to generate frequent itemsets [15].

WANG Guo-Yin developed a method for calculating the core attributes of a decision table [16].

Wanjun Yu, Xiaochun Wang have proposed a novel algorithm called as Reduced Apriori Algorithm with Tag (RAAT), which improves the performance of Apriori algorithm by reducing the number of frequent itemset generated in pruning operation, by applying transaction tag method [17].

Wanjun Yu, Xiaochun Wang developed a method to find more valuable rules [18].

XIAO Bo et al. proposed a MaxCliqueMining algorithm which creates 2-item credible sets by adjacency matrix and then generates all rules based on maximum clique [19].

III. APRIORI ALGORITHM

Apriori algorithm employs an iterative approach known as level-wise search, where k-item sets are used to explore (k+1)- item sets. First, the set of frequent 1-itemsets L1 is found. Next, L1 is used find the set of frequent 2-itemsets L2. Then L2 is used to find the set of frequent 3-itemsets L3.The method iterates like this till no more frequent k-item sets are found.

Apriori Algorithm for FI

```
Initialize: k := 1, C1 = all the 1- item sets;
read the database to count the support of C1 to
determine L1.
$L_1$ := {frequent 1- item sets}; k:=2;
                    //k represents the pass number//
while ($L_{k-1} \neq \emptyset$) do
begin
     $C_k$ := gen_candidate_itemsets with
                    the given $L_{k-1}$
     prune($C_k$)
for all transactions t € T do increment the count of all
candidates in $C_K$ that are contained in t;
     $L_k$ := All candidates in $C_k$ with
     minimum support ;
     k := k + 1;
end
Answer := $U_k L_k$
```

IV. FUZZY ASSOCIATION RULES AND FUZZY MEASURES

Boolean association rules can be generalized to be fuzzy association rules by representing association between two fuzzy item sets instead of two crisp item sets. In order to process the extended fuzzy dataset, we need new measures which are in terms of t-norms. The more generally used t-norms are listed in Table 1.

Table 1 Fuzzy t-norms in Fuzzy Sets

$T_M(x, y) = \min(x, y)$
$T_P(x, y) = xy$
$T_W(x, y) = \max(x + y - 1, 0)$

V. PROPOSED METHOD

The proposed algorithm works this way. The algorithm converts the crisp dataset into fuzzy dataset and the data is compressed in the form of a matrix. The frequent patterns are then mined directly from this matrix. A new transaction reduction and support count techniques are designed and performed on matrix to achieve efficiency.

Based on the current study; a novel combination of features like fuzzy sets and a new count based transaction reduction method contributes a lot of efficiency in performance which is described as follows.

A. Conversion of Information System into a Fuzzy Item Set Matrix Strategy

Property 1 Construct every candidate k-itemset, I_k, as a fuzzy set on set of qualified transactions, M.

A fuzzy membership function, μ, is a mapping:

$\mu_{I_k}: M \rightarrow [0,1]$ as defined by:

$$\mu_{I_K}(T) = \inf_{i \varepsilon I_K} \left\{ \frac{\eta_T(i)}{card\ (t)} \right\}, \forall\ T \varepsilon M \tag{1}$$

where Ik ⊆ $\mathfrak{I}$; T be a qualified transaction in which T can be regarded also as a subset of items (T ⊆ $\mathfrak{I}$).

B. New Transaction Reduction Strategy

Property 2 Each value of RC column stores the corresponding number of similar rows. If the transaction doesn't repeat then repetition column for the transaction is set to 1.

$$RC_i = \begin{cases} RC_i + 1, if T_{sj} = T_{tj}(s \# t), \\ 1, otherwise,\ j = 1,2,..,n.i, s, t = 1,2,...,m \end{cases} \tag{2}$$

C. New Support Count Strategy

Property 3 Support count of one item set is calculated as counting each column based on its fuzzy value greater than 0.0.

$$sum = \sum_{i=1}^{m} T_{ij},\ j = 1,2,..,n; T_{ij} > 0 \tag{3}$$

Based on that count, the rows are sorted in ascending order. The major advantage of this approach is that, the number of transactions to be scanned is greatly reduced, by reducing the number of similar rows as well as by cutting down the unnecessary transaction rows. So, the corresponding item set is extracted directly without moving for entire database.

Property 4. The support for every candidate k –item set can be got by using value in the RC column and fuzzy t-norms namely intersection.

$$\sup - count - of - k - itemsets = \sum_{i=1}^{n}\sum_{j=1}^{k} RCAx(\min(T_{ij})) \tag{4}$$

VI. WORKED EXAMPLE

In general, a transactional database consists of a file in which each record represents a list of items purchased in a transaction. Simply, a transaction includes a unique transaction identity number and the list of items making up the transaction. The process is started from a given transactional database as shown in Table 5.1.

Here the $\min_{sup}$ is considered as 2. First scan the database to find the different items occurring in the database and then make the matrix by writing all the transactions along the row side and all the items occurring in the database along the column side as shown in Figure 5.1.

Table 5.1: Transactional Database

TID	ITEMS
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

	I1	I2	I3	I4	I5
I1,I2,I5	1	1	0	0	1
I2,I4	0	1	0	1	0
I2,I3	0	1	1	0	0
I1,I2,I4	1	1	0	1	0
I1,I3	1	0	1	0	0
I2,I3	0	1	1	0	0
I1,I3	1	0	1	0	0
I1,I2,I3,I5	1	1	1	0	1
I1,I2,I3	1	1	1	0	0

Figure 5.1: Bit Array Matrix

The fuzzy item sets are constructed as shown in Figure 5.2.

	I1	I2	I3	I4	I5	RCA
I1,I2,I5	0.33	0.33	0.00	0.00	0.33	Null
I2,I4	0.00	0.5	0.00	0.5	0.00	Null
I2,I3	0.00	0.5	0.5	0.00	0.00	Null
I1,I2,I4	0.33	0.33	0.00	0.33	0.00	Null
I1,I3	0.5	0.00	0.5	0.00	0.00	Null
I2,I3	0.00	0.5	0.5	0.00	0.00	Null
I1,I3	0.5	0.00	0.5	0.00	0.00	Null
I1,I2,I3,I5	0.25	0.25	0.25	0.00	0.25	Null
I1,I2,I3	0.33	0.33	0.33	0.00	0.00	Null
SUM	**6**	**7**	**6**	**2**	**2**	

Figure 5.2: Fuzzy Set Matrix

1. The support count of one itemset is calculated as counting each one itemset based on its fuzzy value greater than 0.0. Move all those transactions to L_1 whose sum value is not less than min_support ($\min_{sup}$=2). Therefore $L_1 = \{I_1, I_2, I_3, I_4, I_5\}$.

2. By using equation 2, each value of RCA column stores the corresponding number of similar rows. If the transaction doesn't repeat, then a repetition column for the transaction is set to 1 as shown in Figure 5.3.

Now for the generation of two item sets, consider the matrix again. Calculate the support count for 2 - itemsets by using equation 4.

sup_count(I_1,I_2)=4 sup_count($I_{1.}$ I_3)=4
sup_count(I_1,I_4)=1 sup_count(I_1,I_5)= 2
sup_count($I_{2,}$ I_3)=4 sup_count($I_{2,}$ I_4)=2
sup_count($I_{2,}$ I_5)=2 sup_count(I_3, I_5)=2

I1	I2	I3	I4	I5	RCA
0.33	0.33	0.00	0.00	0.33	1
0.00	0.5	0.00	0.5	0.00	1
0.33	0.33	0.00	0.33	0.00	1
0.5	0.00	0.5	0.00	0.00	2
0.00	0.5	0.5	0.00	0.00	2
0.25	0.25	0.25	0.00	0.25	1
0.33	0.33	0.33	0.00	0.00	1

Figure 5.3: Transaction Reduction using RC

Then move only those item sets L_2 whose support count value is not less than minimum support. Therefore, $\{I_{1,}\ I_{2}\}$,$\{I_{1,}\ I_{3}\}_{,}$ $\{I_{1,}\ I_{5}\}_{,}$$\{\ I_{2,}\ I_{3}\}_{,}$ $\{I_{2,}\ I_{4}\}_{,}$ $\{I_{2,}\ I_{5}\}_{,}$ $\{I_{3,}\ I_{5}\}$ will be frequent 2 - itemsets.

4. Next, consider all the 3-itemsets combinations of the items the various combinations possible are $\{I_1, I_2, I_5\}$; $\{I_1, I_2, I_4\}$; $\{I_1, I_2, I_3\}$; $\{I_2, I_3, I_5\}$; $\{I_1, I_3, I_5\}$.

Now, by using equation 4, sup_count((I_1, I_2, I_5) =2 sup_count((I_1, I_2, I_3) =2. Therefore, $\{I_1, I_2, I_5\}$ & $\{I_1, I_2, I_3\}$ will be the collection of 3 itemsets.

5. The proposed algorithm is terminated because there are two frequent 3-itemsets in the set of frequent 3-itemset L_3. Therefore, the frequent 4 itemsets does not exist.

In equation 4, we combined the fuzzy intersection and value of RC column for the new support count. In fuzzy intersection, if any one item has a value 0, then the intersection is not continued for the remaining items in a transaction. RC contains the count of the similar transactions; it can greatly reduce the number of transactions to be scanned. So, when we combine the above said features, it further reduces the time for finding frequent itemset when compared to the above mentioned Apriori algorithm.

VII. PERFORMANCE ANALYSIS

In order to appraise the performance of the proposed algorithm, we conducted an experiment using the Apriori algorithm and the proposed algorithm.

A. Experiment 1

For this purpose, we select A.S Hospital, Ariyalur data to study the object. The Hospital Information System contains 1000 records and this dataset contains 15 attributes as shown in Table 5; apply algorithms on same number of record and compare the execution time with support count 5, 10,15,20,25 and it is shown in Figure 4.

Table 5 Attributes Information

ID	Attribute	ID	Attribute
1	angina	8	Swollen- Feet
2	Throat or Jaw pain	9	Swollen- Ankles
3	Irregular Heart Beat	10	Heart Pain
4	Shortness of Breath	11	Stomach Pain
5	Pain - Arm	12	Sweating
6	Dizzy	13	Snoring
7	Fever	14	Swollen –Legs
		15	Cough

Figure 4 shows performance of two algorithms; here proposed algorithm performs better in order to time efficiency.

The time reducing rate is calculated by

$$TRR = ((existing\_a\lg(ms) - proposed\_a\lg(ms))/existing\_a\lg(ms))X100$$

As we observe in Figure 4, that the time consuming in proposed approach in each value of minimum support is less than it in the original Apriori, and the difference increases more and more as the value of minimum support decreases.

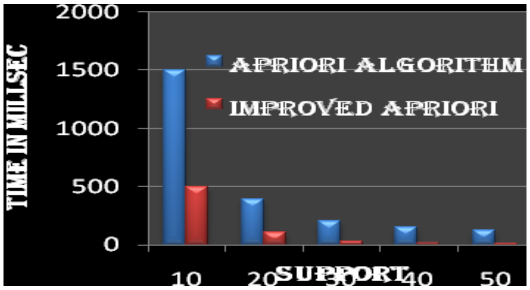

Figure 4 Time Consuming Comparison for Different Values of Minimum Support

In the following Figure 5, we show the number of candidates generated in proposed method less than it in the original Apriori.

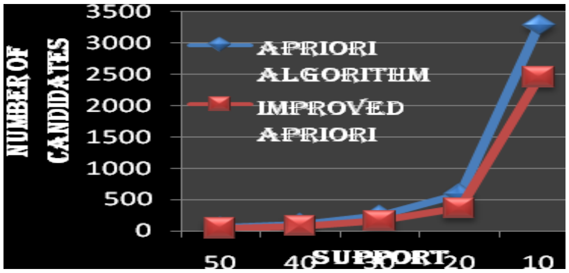

Figure 5 Number of Candidate Itemsets Comparison

B. Experiment 2

The second experiment compares the time consumed of original Apriori and our Proposed Algorithm by applying the five groups of transactions in the implementation. For this purpose we select datasets from [20]. The result is shown in Figure 6 and Figure 7.

- T1 : 958 Transactions
- T2 : 999 Transactions
- T3 : 1000 Transactions
- T4 : 1200 Transactions
- T5 : 1750 Transactions

a. Time Complexity Analyses:
The Improved Apriori algorithm only needs to scan the database for one time, avoiding the connection and pruning operation.

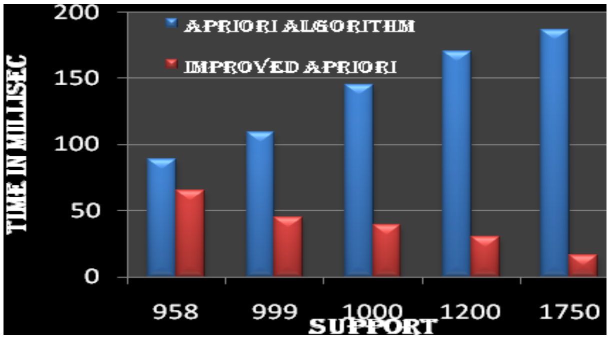

Figure 6 Time Consuming Comparison for Different Group of Transactions

b. Space Complexity Analyses:
The data of Apriori algorithm stored is the item value, and in the process of solving the frequent itemsets, it takes up a large amount of space. The proposed method just scans the information table only once, and at the same time filters unexpected frequent sets through support count, reduces the number of candidate itemsets. As a result it has saved storage space.

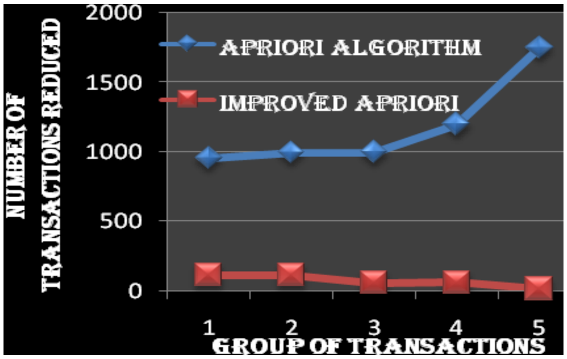

Figure 7 Number of Transactions Reduced Comparison for Different Group of Transactions

The compressed ratio between the Apriori and Improved Apriori algorithm for support factor of 15% and for different data sets is presented in Table 6.

Table 6 Compression Ratio

Min_Sup	Apriori	Improved Apriori	Compressed Ratio (%)
15%	958	116	87.8
	999	116	88.3
	1000	61	93.9
	1200	63	94.75
	1750	16	96

VIII. IMPLEMENTATION

All experiments were performed on Intel core i3, 3.07GHz processor and 2GB of RAM, the algorithms were implemented in Java and tested on a Windows XP platform.

IX. CONCLUSION

This paper has studied the classic association rules mining algorithm and discussed shortcomings of the Apriori algorithm. It has designed and implemented the proposed algorithm, and an existing medical details gained from hospital are used as training data set for data analysis. The outcome of the study is that this algorithm can be efficiently used to discover frequent diseases in a large medical dataset and it will help the practitioners in making medical decisions for frequently occurring diseases. The paper has proved theoretically and experimentally that this attribute-transaction reduction method is one of the best frequent pattern mining methods.

REFERENCES

[1] R.Agrawal and Srikant. R "Fast Algorithms for Mining Association Rules". In: Proceedings of 20th International Conference of Very Large Data Bases. pp. 487-499, 1994.

[2] J. Ayres and Flannick.J, Gehrke.J, and Yiu.T " Sequential pattern mining using a bitmap representation" .In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 429-435 , 2002.

[3] Changsheng Zhang and Jing Ruan "A Modified Apriori Algorithm with its application in Instituting Cross-Selling strategies of the Retail Industry". In: Proc. of International Conference on Electronic Commerce and Business Intelligence. pp. 515-518, 2009.

[4] J. Chen and Xiao.K " BISC: A bitmap itemset support counting approach for efficient frequent itemset mining". In: ACM Transactions on Knowledge Discovery from Data (TKDD). vol. 4, p. 12, 2010.

[5] Chen Chu-xiang, Shen Jian-jing, Chen Bing, Shang Chang-xing, Wang Yun-cheng "An Improvement Apriori Arithmetic

based on Rough set Theory" .In: Third Pacific-Asia Conference on Circuits, Communications and System (PACCS). pp.1 - 3 , 2011.

[6] Dongme Sun and Sheohue Teng "An algorithm to improve the effectiveness of Apriori Algorithm". In: Proc.of 6th ICE Int. Conf. on Cognitive Informatics. pp. 385-390, 2007.

[7] Feldman, Aumann.R, Amir.Y, Zilberstain.A, Kloesgen.A, Ben-Yehuda (W), Maximal association.Y "Association rules: a new tool for mining for keyword co-occurrences in document collection". In: Proceedings of the 3rd International Conference on Knowledge Discovery. pp. 167-170, 1997.

[8] J.W.Guan, Bell, D.A, Liu, D.Y "The Rough Set Approach to Association Rule Mining". In: Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03). pp.529 – 532,2003.

[9] Hanbing Liu and Baisheng Wang "An Association Rule Mining Algorithm Based on a Boolean Matrix" In: Data Science Journal. Vol.6, supplement 9,2010.

[10] Jaisree singh, Hari Ram, Dr.J.S. "Improving efficiency of Apriori Algorithm using Transaction Reduction". In: IJSRP. Vol. 3. ISSN 2250-3153, 2013.

[11] K.Kavitha.and Dr.E.Ramaraj) "Efficient Transaction Reduction in Actionable Pattern Mining for High Voluminous Datasets based on Bitmap and Class Labels". In: IJCSE. Vol.5 No.07 Jul 2013. ISSN: 0975- 3397,2013.

[12] T.Logeswari, Valarmathi.N, Sangeetha. A, Masilamani.M " Analysis of Traditional and Enhanced Apriori Algorithm in Association Rule Mining" . In: International Journal of Computer Applications. Vol.87, 2013.

[13] A. Pethalakshmi and V.Vijayalakshmi "An Efficient Count Based Transaction Reduction Approach For Mining Frequent Patterns". In: Procedia Computer Science- Elsevier,Vol.47, pp. 52-61,2015.

[14] E.Ramaraj, K.RameshKumar, N.Venkatesan "A Better Performed Transaction Reduction Algorithm for Mining Frequent Itemsets from Large Voluminuous Database". In: Proceeding of the 2nd National Conference, Computing for Nation Development, February 08, 2008.

[15] Sixue Bai and Xinxi Dai "An efficiency Apriori algorithm: P_matrix algorithm". In: First International Symposium on Data, Privacy and Ecommerce. pp.101-103, 2007.

[16] WANG Guo-Yin " Calculation Methods for Core Attributes of Decision Table[J]". In: CHINESE JOURNAL OF COMPUTERS. Vol. 26(5): 611-615, 2003.

[17] Wanjun Yu and Xiaochun Wang "The Research of Improved Apriori Algorithm for Mining Association Rules". In: Proc. of 11th IEEE International Conference on Communication Technology Proceedings. pp. 513-516, 2004.

[18] XIA Ying, ZHANG WANG Guo-yin. "Spatio-temporal Association Rule Mining Algorithm and its Application in Intelligent Transportation System[J]". In: Computer Science. Vol. 38(9): 173-176, 2011.

[19] XIAO Bo, XU Qian-Fang, LIN Zhi-Qing, GUO Jun, LI Chun-Guang "Credible Association Rule and Its Mining Algorithm Based on Maximum Clique[J]". In: Journal of Software. (10): 2597-2610, 2018.

[20] Vaibhav Jain, "Evaluating and Summarizing Students's Feed Back Using Opinion Mining", International Journal of Scientific Research in Computer Science and Engineering. Vol.1.Issue 1. Jan-Feb 2013.

[21] Vaibhav Jain, "Frequent Navigation Pattern Mining from Web usage ", International Journal of Scientific Research in Computer Science and Engineering. Vol.1.Issue 1. Jan-Feb 2013.

Authors Profile

Mrs. V. Vijayalakshmi received her Master of Computer Application from Bharathidasan University, Tamilnadu, India in 1999 and the Master of Philosophy in Computer Science from Mother Teresa Women's University, Kodaikanal, Tamilnadu, India in 2003. She obtained her Ph.D degree in Computer Applications from the Manonmaniam Sundaranar University, Tirunelveli in 2017. Currently she is working as Assistant Professor, PG & Research Department of Computer Science, Government Arts College, Ariyalur. Her research interest includes Data Mining, Fuzzy Set and Rough Set