Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/130604

An Efficient Algorithm for Mining Frequent Itemsets in Large Databases

B. Praveen Kumar[1,*] *and D. Paulraj*[2]

[1] Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, Tamilnadu, India
[2] Department of Computer Science and Engineering, R.M.D. Engineering College, Tamilnadu, India

Received: 2 Apr. 2019, Revised: 2 May 2019, Accepted: 12 May 2019
Published online: 1 Nov. 2019

Abstract: Flexible Bitmap Index and Cluster based Bit Vector (FBICBV), focus on identifying the eligible candidates from unfiltered huge volume of temporal data in order to find out frequent patterns among them. One of the best and efficient solutions is to use the mechanism of bitmap indexing and clustering. First, using bitmap index, the rich data are filtered out from the unfiltered raw dataset to be analyzed effectively. Thus, the eligible candidate data are identified through this process. Second, the frequent patterns are identified using cluster based bit vector appropriate for effective decision making. Hence, scanning of raw data is completely avoided using bitmap index. Also, it eliminates the storage of the candidate's unnecessary data while forming a cluster table. Consequently, it implies improvements in optimizing the database storage for maximum performance and efficiency using FBICBV algorithm compared existing algorithms.

Keywords: Frequent Item sets, Bitmap Index, Bit Vector, Clustering, Temporal Data.

1 Introduction

Data mining is valid, potential and meaningful process and ultimately discovers data. It enables you to discover new intelligence that hides your data. In addition, it is a fast-evolving area that includes temporal mining, statistics and analytics, template identification, optimization as well as high performance computing. The present paper aims to identify the exact data of the data mining information derived from the patient's medical and clinical data.

1.1 Analysis in Frequent Itemset

Market basket analysis aimed to find the itemset to properly locate supermarkets in product analysis. In particular, itemsets or frequent forms are identified using data mining. In the continuous itemset mining industry [1], the information takes place in each of the various commodity transactions boxes. At each cycle of length n, length $1 - n$ itemsets are produced. The items need to be scanned repeatedly and huge candidates are generated. This is a major disadvantage of the predatory process thus it is necessary to propose various methods that provide rapid results and thereby solve this problem.

2 Partitioned Clustering

Clustering is categorizing data into groups with similar objects. Data mining adds to complexities of clustering a large dataset with various features in temporal data. Clustering data is a technology that offers modelling data as a summary. Clustering plays a vital role in a several applications related to various sectors. This paper comprises clustering information and the knowledge diagnostic technique for the discovery of the patient's medical insurance application.

2.1 Temporal Data Mining

Temporal data mining manages a lot of information related to time. Human activities and their associations in biomedical and social domain have expanded the significance of temporal mining in the world of information.

Several creations analyze the shape of the association, temporal mine and frequent itemsets. Data is a fundamental experiment in the mining industry. Computations [1, 2] create important item association information and scanning that require computations.

* Corresponding author e-mail: praveenv1754@yahoo.com

rules. However, candidate itemsets give birth to large amounts of

Many studies were conducted to minimize the database filtering and further studies in knowledge mining have presented numerous efficient algorithms that help find association rules. In the improved apriori calculation [3], mining efficiency is extremely inadmissible when the memory for database is considered. Unique pattern frequency systems still have no common prefixes within the knowledge of the accounting data [4, 5]. Before their development, the previously created candidate surpassed all the items.

Another method [6] of temporal shutter mining is proposed to frequently detect the theoretical method in clustering, bit vector and variable threshold view. Along with the continuous forms and non-standard shapes, the forms are characterized by another type, i.e. mixed type [7]. It helps the retailer make its own methodology according to the requirement of time. However, its performance is very low.

The itemsets processing procedure [8] proposed a work that was used in conjunction with the subsequent guidelines on globalization. It detected that all potential transactions are stamped on time. The vector bit algorithm for processing is often based on the findings of the Cluster Globalized Databases (CBVAR). Use information for a single scan towards the generic itemsets generation. The ideas [5, 7] have minimized itemsets, scanning loss and execution times, although the procedure operates for nine transactions. The sharing algorithm [8] is an improved algorithm, which has reduced the number of information scans and spent more time to scanning inferior candidate itemsets [8, 9].

In an efficient Boolean algorithm for mining association rules in large databases, logical AND operations are utilized to compute frequent itemsets [10]. In the second step, logical AND and XOR operations are connected to define all interesting association rules in light of the registered frequent itemsets. However, the computational time is more and additionally it involves more memory. An association clause, where clustering and map are located [11], is a cluster of binary-based transactions in the view of their length [12], which is indicated when a database is presented in a bunch table.

There is a need for computation matching and its changes require some information to be investigated. The strategy for putting close essential data, and another database architecture, frequently proposed tree (federal party-tree), requires only two database checks in this strategy when digging all frequent itemsets in the mining [13, 14].

3 Limitations of Existing Work

An enhanced algorithm was proposed to produce frequent k-itemsets without new candidate generation. The change is primarily to decrease query frequencies and storage resources. In the case of thick database, there will be N-itemsets. First, the database is filtered. This method comprises two stages: join step (Combining every item with the other items) and prune step (filtering). If the support is less than the minimum support figure, the candidate will shrink and reverse the products. One of the calculation limits involves the forearm of memory storage space. Thus, as long as the itemsets rapidly update, the information is quicker when the itemset algorithm is in the TIMV repair mode (FUFIA) and when the path we are picking is visible to the itemset matrix need to fix the essential itemsets when the table team is change.

If new information is included, we consider it a part of important information. Therefore, the main itemsets team should be adjusted. The bit vector length can be expanded. We want to erase the possibility that the value will be decreased by the expansion on the day M [ii, ij, ik], and the length will be reduced to the vector. In FUFIA, the procedure involves some of the most complex space and protocols, and some transactions may continue to be present.

In fact, CGAR has often suggested using the data structure to have the ability to transform into Q-itemsets, as k times = 2. CGAR marks a list of items that are present to indicate columns and rows of transactions where a bunch of data sets in a two dimensional array and checks the information. The tuples are binary. The work schedules of the Boolean association rules should be built in every bunch of dimensional line approach and have their own control. Therefore, such problems have been suggested by clustering and bit vector feedback. The issue of mining continuous item set is locating the complete arrangement of frequent itemset in a given transaction database.

An itemset is regular if its support is more than any user's defined minimum support threshold. The main data structure of the bit vector mining calculations is the matrix. In the second calculation, a variant of the matrix that gives the best result in finding frequent itemset called Three-dimensional Itemset Matrix (TDIM), is provided. CBVAR filters the database of transactions just once to construct the clustering table as a 2-D array, where the columns denote items and the rows denote Transaction Identifiers (TID) [15].

CBVAR schemes and arrays are sorted by transmitting 2-D sequence information, such as package table (Terrorism Investigation Division) where items are indicated to denote the transaction ID. The bits (0 or 1) in the table represent a presence or absence of an object, respectively. now, the masonry table contains information for each individual bit of information that is not dummy material, but it is evaluated as CBVAR wipe. In this way, it is important to locate the itemsets to use the technique as an alternative. Hence, Enhanced Vector Cluster Located Association Cluster Area (ECBVAR) is CBVAR exaggeration.

CBVAR is the same as the ECBVAR method, but it uses quantitative comparisons to quantify the transaction

number (CNT in the items). 0 or 1 bits in the table indicate absence or presence, respectively. Each section represents the number of occurrences of an object in the 1's database. The event transaction method (events) explains the same transfers as the same transactions. So, the single transaction is also linked to a number that will be updated according to the number of the value. After increasing the count, the duplicate transactions are erased except the transaction that has updated the count value. The measure of memory consumed by the database is decreases here; it is improvement of CBVAR. Although it is better than CBVAR, the execution time of ECBVAR is more comparing with FBICBV (Flexible Bitmap Index and Cluster based bit vector association rule mining) because the duplication value is erased after updating the table. Consequently, the above-mentioned limitations motivate me to propose a new algorithm called FBICBV.

4 Proposed Work

The proposed algorithm makes it possible to mine the unfiltered temporal data to identify the eligible candidates based on the frequent itemset mining preferred as association rule. Furthermore, it enables us to configure the business rules according to the business needs. The business rules are applied using bitmap indexes. Thus it helps avoid the unnecessary data scans at the very initial stage using bitmap indexes. Subsequently, it uses clustering and bit vector logic to identify the frequent patterns among the identified eligible candidates. Hence, its main advantage is avoiding unnecessary data scans, reduced storage requirements using bitmap indexes, and dramatic performance gain because bitmap indexes are most effective for queries that contain multiple conditions in which clauses are based on business rules. Various algorithms were proposed to increase the performance and to reduce the space. However, most of them not remove the unnecessary data and thereby occupy more space and generate many candidate sets.

In order to address this issue, the new algorithm Flexible Bitmap Index and Cluster based Bit Vector association rule mining (FBICBV) has been proposed. It significantly reduces the scanning cost by removing the unnecessary data in advance. Moreover, it increases performance and reduces the storage requirement. This strategy utilizes bitmap lists which involve several valuable properties. For instance, the count and the bit-wise operations can be utilized through selection queries. Our strategy has two noteworthy points; it is not restricted by the size of the main memory and it enhances preparing time because there is no mandatory reason to access information sources because our method utilizes bitmap record rather than entire training set. A bitmap record is information structure used to productively access huge databases.

The inspiration behind a bitmap index is to set up a bit either to 1 or 0 to an attribute in a table containing given key qualities. In our example, if the bit is set to "1", it infers that the attribute contains the key quality; otherwise, the bit is set to "0".

Algorithm 1: Proposed Algorithm

Input : Dynamic Rules Table R, Temporal Unfiltered Database D
Output : Frequent Item sets
Procedure FBIBV(R:D). Table 1, Temporal Unfiltered Database D
Create Index on Temporal Unfiltered Database D on Dynamic Rules Table R (Attribute1, Attribute2)
Form a Filtered Table F from Temporal Unfiltered Database D based on the Index Value
Form a Cluster Table from the given Filtered Table
Decide frequent k-itemsets
Frequency count = 0
Support threshold = 0
$k = 1$
L_k: incessant itemset of size k
C_k: applicant itemset of size k
CNT: Transaction Count
Consolidate the identical rows into one row and augmentation CNT appropriately generate candidate itemset, Ck by seeing the 1's in the comparing item position.
If $k == 1$ then
Increment i loop from 1 to NT times
Increment i loop from 1 to NI times
In the event that Column $bit[j] == 1$
Go to loop
else
Till every one of the lines are one of a kind then
Increment i loop from 1 to NT times
j loop from 1 to NI times
If (Column $bit[j]$ & & Column $bit[j+1])jj$ Column $bit[j]$ jj Column $bit[j+2]$ jj Column $bit[j]$ & & Column $bit[j+3]$ $jj \ldots jj$ Column $bit[j]$ & & Column $bit[j+NI]$) then
Creates each one of the subsets which are recognized with the given itemset
Implement i loop from 1 to NT times
Frequency count = frequency count + CNT
Support threshold = frequency count * NI
Increment j from 1 to NI times
If Support threshold > minthreshold[j]) then
Print the frequent itemset
Return L_k
Reset frequency count and Support threshold
else
Erase that item(s)
Reset frequency count and Support threshold
end

Table 1: Dynamic rules table

	Index column	
Bitmap Index	1	0
Payer	PAYER1	Other
Age	≥ 34	< 34

4.1 Implementation of FBICBV

The FBICBV algorithm uses bitmap index that makes it possible to mine the unfiltered data in order to define the eligible candidates. Referred as association rule, bit map index is set to 1 or 0 on the key columns which is used to take an important decision. Setting up the bit map index can also be altered dynamically in the future based on the business needs. Thus, the eligible candidates can be identified by altering the bit map indexed column. The FBICBV algorithm has been implemented to identify the patients who have a high risk factor for heart disease and heart attack according to the census file received from different payers or insurance organizations.

In medical data, the patients, covered under PAYER 1 of the age group above 34, are identified using bit map index in order to find the patients who have high risk factor for heart disease and heart attack. With that context, the bit map index Table 1 is formed as shown below. Table 1 consists of two attributes: payer and age. Bit map index is set to 1 if the payer's attribute value contains "Payer1 and 0 if the payers attribute value contains "Other and similarly, for age. Bit index is set to 1 if the patient's age is ≥ 34 and 0 if his/her age is < 34.

This bit map indexing can also be conducted based on gender as future enhancement through adding one more index on attribute gender. Bit index will be set to 1, for example, for male and 0 for female. Thus, the bit map index can be changed dynamically based on the business rules or business needs.

The patient census files received from payers are stored in Table 2, which contains assessment details of the patients with attributes Payer, Insurance ID, First Name, Last Name, Gender, Age, Health Studies and Assessment Conducted Date.

Now, create index on temporal unfiltered database based on Table 1 R (Payer, Age). Accordint to Table 1, any records with PAYER 1 insurance will be set to 1 and other insurance records are set to 0. The same logic is applied to age attribute are as well. Records in which age ≥ 34 are set to 1 and the other records in which age < 34 is set to 0. Based on these rules, Table 3 is created.

From the above mentioned unfiltered patient statistics data, the below filtered data containing patient details of PAYER 1 members of age ≥ 34 are derived using bit map indexing. The patient details that have insurance as 'Other' are removed and patient details of PAYER 1 members of age < 34 are also eliminated because these data are considered as ineligible data.

Total number of eligible candidates (Patient details of PAYER 1 of age ≥ 34) = 28 and Total No. of ineligible candidates (Patient details of other Payer or PAYER 1 of age < 34) = 12.

Accordingly, total of 12 records are eliminated using Bitmap index. In reality, the volume of medical data will be huge. So using Bit map indexing, the eligible candidate sets are identified for further processing.

The cluster Table 5 is formed using bit vectors. The bit vectors for the symptoms tobacco, HBP (High Blood Pressure), LDL (Cholesterol), Diabetes and Stress are represented as follows: if patient has the symptoms, the corresponding value is denoted as 1. If he/ she does not have them, it is set to 0. Also, using clustering, 3 clusters are formed for the patients observed with presence of two symptoms, patients observed with two, three and four symptoms.

In Table 6, if one or more patients are observed with similar health problem or symptoms, the assessment transaction details are clubbed together by incrementing the count value to avoid duplicate patterns. In Table 5, Assessment No. 1,4 and 3, 5 have similar bit representation suggesting that both patients have similar symptoms and thereby all the three rows are combined and the count value is represented as 3 in Table 6. As a result, the storage space is also reduced.

Bit vectors for the Symptoms Tobacco, HBP, LDL, Diabetes and Stress are given below:

$$BV(\text{Tobacco}) = 01101000100111011111111$$
$$BV(\text{HBP}) = 10000001000000011001001$$
$$BV(\text{LDL}) = 10110111011011000100111$$
$$BV(\text{Diabetes}) = 00001010101000111111111$$
$$BV(\text{Stress}) = 01010100001010110100110$$

Suppose the minimum support threshold is 50%. Support threshold for the symptoms observed is frequency of the symptom × total number of symptoms.

$$BV(\text{Tobacco}) = 14 * 5\% = 70\%$$
$$BV(\text{HBP}) = 6 * 5\% = 30\%$$
$$BV(\text{LDL}) = 13 * 5\% = 65\%$$
$$BV(\text{Diabetes}) = 12 * 5 = 60\%$$
$$BV(\text{Stress}) = 10 * 5\% = 50\%$$

The support threshold of HBP symptom is less than 50%, So it is removed from the database. The other symptoms (Tobacco, LDL, Diabetes, and Stress) are considered because the support threshold is greater than or equal to 50%. In Table 7, the similar styles of the signs and symptoms are collectively combined to put off repetitive assessment transactions.

Below are the bit vectors from Table 7.

$$BV(\text{Tobacco, Stress}) = 8 * 5 = 40\%$$

Table 2: Temporal unfiltered master table

Assess. No	Payer	Ins. ID	First Name	Last Name	Gender	Age	Health Study	Assess. Date
1	PAYER 1	111	Tina	Jacob	F	35	Tobacco, LDL, Diabetes, Stress	1/12/2017
2	PAYER 1	112	Jackson	Ted	M	35	HBP, LDL	1/12/2017
3	PAYER 1	113	Aiden	Tony	M	50	Tobacco, Stress	1/12/2017
4	PAYER 1	114	Alphy	Shawn	F	45	Tobacco, LDL, Diabetes, Stress	2/12/2017
5	PAYER 1	115	Sylvia	Aaron	F	55	Tobacco, LDL	3/12/2017
6	PAYER 1	116	Liam	Peter	M	60	Tobacco, LDL, Stress	10/12/2017
7	PAYER 1	117	Elizabeth	Daniel	F	74	LDL, Stress	10/12/2017
8	PAYER 1	118	Jenifer	Alex	F	68	HBP, Diabetes, Stress	10/12/2017
9	PAYER 1	119	Lucas	Philip	M	75	Tobacco, HBP, LDL, Diabetes	14/12/2017
10	PAYER 1	120	Rebecca	Ryan	F	45	Tobacco, Diabetes	14/12/2017
11	PAYER 1	121	Linda	James	F	55	Tobacco, HBP, Diabetes	15/12/2017
12	PAYER 1	122	Noah	Edwin	M	62	LDL, Stress	18/12/2017
13	PAYER 1	123	Joy	Jack	F	73	Tobacco, HBP, LDL, Diabetes	19/12/2017
14	PAYER 1	124	Mason	Adan	M	58	LDL, Diabetes	19/12/2017
15	PAYER 1	125	Ben	Brian	M	32	HBP, LDL, Diabetes	19/12/2017
16	PAYER 1	126	Emily	Edwin	F	60	Tobacco, Diabetes, Stress	19/12/2017
17	PAYER 1	127	Aswin	Pat	M	38	Tobacco, LDL, Diabetes	22/12/2017
18	PAYER 1	128	William	Eric	M	34	Tobacco, LDL, Diabetes, Stress	22/12/2017
19	PAYER 1	129	George	Dave	M	35	HBP, LDL	22/12/2017
20	OTHER	130	Hannah	Robert	F	50	Tobacco, Stress	22/12/2017
21	OTHER	131	Jessica	John	F	45	Tobacco, LDL, Diabetes, Stress	22/12/2017
22	OTHER	132	Rich	Brook	M	55	Tobacco, LDL	23/12/2017
23	OTHER	133	Albert	Perry	M	60	Tobacco, LDL, Stress	23/12/2017
24	PAYER 1	134	Amy	Aiden	F	34	LDL, Stress	23/12/2017
25	PAYER 1	135	Lucy	Tyler	F	30	HBP, Diabetes, Stress	24/12/2017
26	PAYER 1	136	Steve	Arthur	M	75	Tobacco, HBP, LDL, Diabetes	25/12/2017
27	PAYER 1	137	Lilly	max	F	45	Tobacco, Diabetes	25/12/2017
28	PAYER 1	138	Sophie	Rich	F	55	Tobacco, HBP, Diabetes	25/12/2017
29	PAYER 1	139	Ryan	Carl	M	62	LDL, Stress	26/12/2017
30	OTHER	140	Grace	Mithun	F	73	Tobacco, HBP, LDL, Diabetes	26/12/2017
31	PAYER 1	141	Kelly	Kevin	F	58	LDL, Diabetes	26/12/2017
32	PAYER 1	142	Mithun	Cruz	M	32	HBP, LDL, Diabetes	27/12/2017
33	PAYER 1	143	Jasmine	Luke	F	60	Tobacco, Diabetes, Stress	27/12/2017
34	OTHER	144	Andrew	Donald	M	38	Tobacco, LDL, Diabetes	27/12/2017
35	PAYER 1	145	Bob	David	M	62	Tobacco, LDL, Diabetes, Stress	28/12/2017
36	OTHER	146	Ella	Harry	F	73	HBP, LDL	28/12/2017
37	PAYER 1	147	Bobby	Davis	M	58	Tobacco, Stress	28/12/2017
38	PAYER 1	148	Sandy	Amir	M	32	Tobacco, LDL, Diabetes, Stress	29/12/2017
39	PAYER 1	149	Sanjay	Eyan	M	60	Tobacco, LDL	30/12/2017
40	OTHER	150	Vanessa	William	F	38	Tobacco, LDL, Stress	31/12/2017

$$BV(\text{Tobacco, LDL}) = 10 * 5 = 50\%$$
$$BV(\text{LDL, Stress}) = 7 * 5 = 35\%$$
$$BV(\text{Tobacco, Diabetes}) = 13 * 5 = 65\%$$
$$BV(\text{LDL, Diabetes}) = 9 * 5 = 45\%$$
$$BV(\text{Diabetes, Stress}) = 6 * 5 = 35\%$$
$$BV(\text{Tobacco, LDL, Stress}) = 4 * 5 = 20\%$$
$$BV(\text{Tobacco, Diabetes, Stress}) = 5 * 5 = 25\%$$
$$BV(\text{Tobacco, LDL, Diabetes}) = 7 * 5 = 35\%$$

Suppose the support threshold for pattern with three symptoms is 25%.

For the patterns with three symptoms are represented in Table 7 for (Tobacco, LDL, Diabetes) their corresponding subsets are also generated. The subset generation includes its subsets {Tobacco, LDL}, {Tobacco, Diabetes}, {LDL, Diabetes} generation. The support threshold of BV (Tobacco, Stress) is computed because it is present in assessment no (3, 37), 6, (16, 33), (4, 1, 35) as $2 + 1 + 2 + 3 = 8 \times 5 = 40\%$. This pattern as support threshold greater than 25% , So it is considered as a high risk factor for heart disease and heart attacks. Hence, the patterns with 3 high risk factors are Tobacco, Diabetes, Stress and Tobacco, LDL, Diabetes. From this data, the preventive care for heart disease and heart

Table 3: Bit map join index table

Assess. no.	Payer	Age	Ins. 1 PAYER 1	Ins. 2 Other	Age 1–< 34	Age 2–≥= 34
1	PAYER 1	35	1	0	0	1
2	PAYER 1	35	1	0	0	1
3	PAYER 1	50	1	0	0	1
4	PAYER 1	45	1	0	0	1
5	PAYER 1	55	1	0	0	1
6	PAYER 1	60	1	0	0	1
7	PAYER 1	74	1	0	0	1
8	PAYER 1	68	1	0	0	1
9	PAYER 1	75	1	0	0	1
10	PAYER 1	45	1	0	0	1
11	PAYER 1	55	1	0	0	1
12	PAYER 1	62	1	0	0	1
13	PAYER 1	73	1	0	0	1
14	PAYER 1	58	1	0	0	1
15	PAYER 1	32	1	0	1	0
16	PAYER 1	60	1	0	0	1
17	PAYER 1	38	1	0	1	0
18	PAYER 1	34	1	0	0	1
19	PAYER 1	35	1	0	1	0
20	OTHER	50	0	1	0	1
21	OTHER	45	0	1	0	1
22	OTHER	55	0	1	0	1
23	OTHER	60	0	1	0	1
24	PAYER 1	34	1	0	0	1
25	PAYER 1	30	1	0	1	0
26	PAYER 1	75	1	0	0	1
27	PAYER 1	45	1	0	0	1
28	PAYER 1	55	1	0	0	1
29	PAYER 1	62	1	0	0	1
30	OTHER	73	0	1	0	1
31	PAYER 1	58	1	0	0	1
32	PAYER 1	32	1	0	1	0
33	PAYER 1	60	1	0	0	1
34	OTHER	38	0	1	0	1
35	PAYER 1	62	1	0	0	1
36	OTHER	73	0	1	0	1
37	PAYER 1	58	1	0	0	1
38	PAYER 1	32	1	0	1	0
39	PAYER 1	60	1	0	0	1
40	OTHER	38	0	1	0	1

attacks can be done for the PAYER 1 patients whose age ≥ 34. Thus, it helps in improving human life.

5 Discussion

Fig. 1 provides a comparison of various values at minimum limits, in CGAR, CBVAR and ECBVAR methods. The list reveals a significant reduction in the time consumption of CBVAR. In addition, the space is similar to the CBVAR prediction algorithm and the same as CGAR. The experimental results demonstrates that the proposed algorithms provide better execution regarding performance and memory space compared with the apriori calculation.

However, the execution time of FBICBV and apriori algorithm are compared. The present paper exhibits that our work is more effective than the other algorithms with

Table 4: Filtered table containing PAYER 1 patients of age ≥ 34

Assess. no.	Health study	Assess. date
1	Tobacco, LDL, Diabetes, Stress	1/12/2017
2	HBP, LDL	1/12/2017
3	Tobacco, Stress	1/12/2017
4	Tobacco, LDL, Diabetes, Stress	2/12/2017
5	Tobacco, LDL	3/12/2017
6	Tobacco, LDL, Stress	10/12/2017
7	LDL, Stress	10/12/2017
8	HBP, Diabetes, Stress	10/12/2017
9	Tobacco, HBP, LDL, Diabetes	14/12/2017
10	Tobacco, Diabetes	14/12/2017
11	Tobacco, HBP, Diabetes	15/12/2017
12	LDL, Stress	18/12/2017
13	Tobacco, HBP, LDL, Diabetes	19/12/2017
14	LDL, Diabetes	19/12/2017
16	Tobacco, Diabetes, Stress	19/12/2017
17	Tobacco, LDL, Diabetes	22/12/2017
19	HBP, LDL	22/12/2017
26	Tobacco, HBP, LDL, Diabetes	25/12/2017
27	Tobacco, Diabetes	25/12/2017
28	Tobacco, HBP, Diabetes	25/12/2017
29	LDL, Stress	26/12/2017
31	LDL, Diabetes	26/12/2017
33	Tobacco, Diabetes, Stress	27/12/2017
35	Tobacco, LDL, Diabetes, Stress	28/12/2017
37	Tobacco, Stress	28/12/2017
39	Tobacco, LDL	30/12/2017

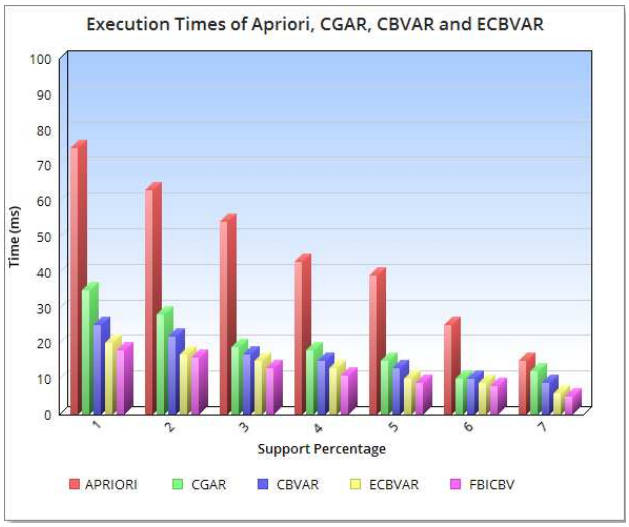

Fig. 1: Execution times of Apriori, CGAR, CBVAR, ECBVAR and FBICBV

Table 5: Cluster table with bit vector values

HS/Assess. no.	Tobacco	HBP	LDL	Diabetes	Stress	Count
2	0	1	1	0	0	1
3	1	0	0	0	1	1
5	1	0	1	0	0	1
7	0	0	1	0	1	1
10	1	0	0	1	0	1
12	0	0	1	0	1	1
14	0	0	1	1	0	1
19	0	1	1	0	0	1
27	1	0	0	1	0	1
29	0	0	1	0	1	1
31	0	0	1	1	0	1
37	1	0	0	0	1	1
39	1	0	1	0	0	1
6	1	0	1	0	1	1
8	0	1	0	1	1	1
11	1	1	0	1	0	1
16	1	0	0	1	1	1
17	1	0	1	1	0	1
28	1	1	0	1	0	1
33	1	0	0	1	1	1
4	1	0	1	1	1	1
9	1	1	1	1	0	1
13	1	1	1	1	0	1
26	1	1	1	1	0	1
1	1	0	1	1	1	1
35	1	0	1	1	1	1

Table 6: Duplicate elimination from Table 5

HS/Assess. no.	Tobacco	HBP	LDL	Diabetes	Stress	Count
2	0	1	1	0	0	1
3	1	0	0	0	1	1
5	1	0	1	0	0	1
7	0	0	1	0	1	1
10	1	0	0	1	0	1
12	0	0	1	0	1	1
14	0	0	1	1	0	1
19	0	1	1	0	0	1
27	1	0	0	1	0	1
29	0	0	1	0	1	1
31	0	0	1	1	0	1
37	1	0	0	0	1	1
39	1	0	1	0	0	1
6	1	0	1	0	1	1
8	0	1	0	1	1	1
11	1	1	0	1	0	1
16	1	0	0	1	1	1
17	1	0	1	1	0	1
28	1	1	0	1	0	1
33	1	0	0	1	1	1
4, 1, 35	1	0	1	1	1	3
9, 13, 26	1	1	1	1	0	3

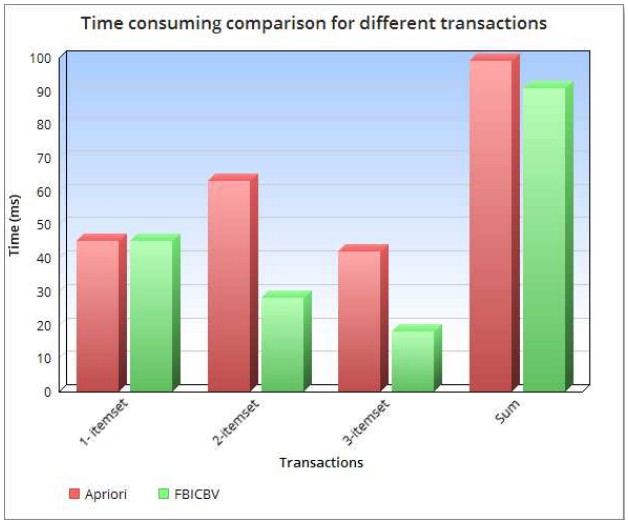

Fig. 2: Time consuming comparison for different transactions

respect to execution time. The proposed algorithm takes less time to form rules comparing with the existing algorithms. Time consuming comparison for different transactions is shown in Fig. 2. Based on the original apriori, our improved version takes less time.

The execution time of 8 items for 10000 transactions and minimum support count is 20%. The execution time of our algorithm is 0.156 seconds and it is less than that of any other existing algorithms.

6 Time Complexity

6.1 Apriori Algorithm

In this strategy, if d items are available in exchange, $2d - 1$ subsets should be produced for which calculation time increments exponentially with the increase in d. This is the principle trap of this strategy.

This calculation will prompt an exponential time unpredictability in most pessimistic scenarios and create $2^d - 1$ subsets which is not ideal.

The apriori algorithm is defence $T = O$ (dn) and compared to the other methods, the FBICBV is less time complexity than other methods.

7 Conclusion

The existing algorithm takes much more space and scans the database even on the ineligible candidate data. Moreover, processing and retrieving the valid data in real world environment play a vital role as we have entered an era of big data. This work finds a solution using FBICBV algorithm in order to extract the valid data from the temporal database of the patient's medical records of

Table 7: Frequent itemset using subset generation

HS/Assess. no.	Tobacco	LDL	Diabetes	Stress	Count	Item Set
2, 19	0	1	0	0	2	NIL
3, 37	1	0	0	1	2	Tobacco, Stress
5, 39	1	1	0	0	2	Tobacco, LDL
7, 12, 29	0	1	0	1	3	LDL, Stress
10, 27, 11, 28	1	0	1	0	4	Tobacco, Diabetes
14, 31	0	1	1	0	2	LDL, Diabetes
8	0	0	1	1	1	Diabetes, Stress
6	1	1	0	1	1	Tobacco, LDL, Stress ⇒ {Tobacco, LDL}, {Tobacco, Stress}, {LDL, Stress}
16, 33	1	0	1	1	2	Tobacco, Diabetes, Stress ⇒ {Tobacco, Diabetes}, {Tobacco, Stress}, {Diabetes, Stress}
17, 9, 13, 26	1	1	1	0	4	Tobacco, LDL, Diabetes ⇒ {Tobacco, LDL}, {Tobacco, Diabetes}, {LDL, Diabetes}
4, 1, 35	1	1	1	1	3	Tobacco, LDL, Diabetes, Stress ⇒ {Tobacco, LDL, Diabetes}, {Tobacco, Diabetes, Stress}, {Tobacco, LDL, Stress}, {LDL, Diabetes, Stress}, {Tobacco, LDL}, {Tobacco, Diabetes}, {Tobacco, Stress}, {LDL, Diabetes}, {LDL, Stress}, {Diabetes, Stress}

insurance system for better analysis compared to other algorithms and to identify the frequent pattern for effective decision making.

The main advantage of this algorithm is using a single scan validation method which is checked to reduce the database selects. The computation time is often too low to take the itemsets generation. This continuous removal often uses fewer steps to produce fake transaction size with patterns. It also helps set bitmap index dynamically through simply adding the business rules in dynamic bitmap table. Thus, the FBICBV reduces space and time, allows flexibility, reduces post computational work and provides better analysis results compared with other algorithms.

References

[1] R. Agarwal, T. Imielinski and A. Swami, Mining Association Rules Between Sets of Items in Large Databases, Proc. ACM SIGMOD Conference on Management of Data, Washington, DC, 207-216 (1993).

[2] R. Agarwal and R. Srikant, Fast Algorithms for Mining Association Rules, Proc. Conference VLDB, 487-499 (1994).

[3] Li Chao and Yu Zhao-ping, Improved Method of Apriori Algorithm based on Matrix, Computer Engineering of China, Vol. 23, 68-69 (2006).

[4] J. Han, J. Pei and Y. Yin, Mining Frequent Patterns without Candidate Generation, Proc. ACM SIGM OD International Conference on Management of Data, New York, ACM press, 1-12 (2000).

[5] Niu Xiao-fei, A High Efficiency Algorithm based on Vectors and Matrix for Mining Association Rules, Chinese Journal of CEA, Vol. 12, 170-173 (2004).

[6] M. Krishnamurthy and A. Kannan, Hybrid Temporal Mining for Finding out Frequent Itemsets in Temporal Databases Using Clustering and Bit Vector Methods, Communications in Computer and Information Science, Springer, Vol. 141, No. 5, 245-255 (2011).

[7] Chaohui Liu and Jianchengan, Fast Mining and Updating Frequent Itemsets, proceedings of ISECS International Colloquium on Computing, Communication, Control and Management, 365-368 (2008).

[8] M. Krishnamurthy, A. Kannan, R. Baskaran and S. Kanmanirajan, Mining Frequent Itemsets using Temporal Association Rule, CiiT International Journal of Data Mining Knowledge Engineering, Vol. 1, No. 1, 40-44 (2009).

[9] Krishnamurthy, Cluster based bit vector mining algorithm for finding frequent itemsets in temporal databases,Procedia Computer Science, Vol. 3, No. 3. 513-523 (2011).

[10] Suh-Ying wur and Yungho Leu, An efficient Boolean algorithm for Mining association Rules in Large Databases, Proc. 6th International Conference on Database Systems for Advanced Applications, 179-187 (1999).

[11] Wael A. Alzoubi, Azuraliza Abu Bakar and Khairuddin Omar, Scalable and Efficient Method for Mining Association Rules, Proc. International Conference on Electrical Engineering and Informatics, 5-10 (2009).

[12] M. Krishnamurthy, A. Kannan, R. Baskaran and M. Kavitha, Cluster based Bit Vector Mining Algorithm for Finding Frequent Itemsets in Temporal Databases, In Journal of Elsevier on Procedia Computer Science, Vol. 3, 513-523 (2011).

[13] Keshri Verma and O.P. Vyas, Efficient Calendar based Temporal Association Rule, SIGMOD Record, Vol. 34, No. 3, 63-70 (2005).

[14] F. Berzal, J.C. Cubero, N. Marin and J.M. Serrano, TBAR: An efficient method for association rule mining in relational Databases, Data and Knowledge Engineering, Vol 37, No. 1, 47-64 (2001).

[15] Yuh-Jiuan Tsay and Jiunn-Yann Chiang, CBAR: An Efficient Method for Mining Association Rules, Knowledge-Based Systems, Vol. 18, No. 2-3, 99-105 (2005).

B. Praveen Kumar received the Bachelor Degree in Electronics and Communication Engineering in (2008). He also received Master Degree in Computer Science and Engineering, from Anna University, Chennai, Tamil Nadu, India. Currently, he serves as Assistant Professor in Computer Science Engineering at Sri Venkateswara College of Engineering, Chennai. His specializations include Data Mining, Big Data as well as Computer Networks and Network Security.

D. Paul Raj received Ph.D. Degree in Computer Science and Engineering, in 2012 from Anna University, Chennai, M.E. in Computer Science and Engineering in 2004 from Anna University, Chennai and B.E. in Computer Science and Engineering in 1993 from Bangalore University, Bangalore. His research areas include Web Technology, Big Data, Cloud Computing, Service Oriented Architecture. Currently he works as a professor at Department of Computer Science and Engineering, R.M.D. Engineering College. Moreover, he has 15 years of teaching experience and 5 years of industrial experience.