ISSN 2250-0987
Venkatesan et al, UNIASCIT, Vol 2 (2), 2012, 263-276

VALIDATION OF ASSOCIATION RULES WITH FINITE AUTOMATA AND HIERARCHICAL CLUSTERING

[1]N. Venkatesan, [2]E. Ramaraj

[1]*Research Scholar, Madurai Kamaraj University, Madurai.*

[2]*Technology Advisor, Madurai Kamaraj University, Madurai*.

Abstract: This paper describes the new approaches for finding the strong and validated association rules for two different applications. Association rule validation is made to avoid irrelevant rules from the newly constructed rules. Association rule analysis starts with transactions containing one or more products or service offerings and some rudimentary information about the transaction. It has two steps process. One is to find the frequent item set from the database. It produce huge quantity of item set, these item set are mostly irrelevant to the transaction. Second step is to construct the association rule from the frequent item set. Two different new validation techniques are introduced and implemented for two different applications. First one is to use Agglomerative hierarchical clustering and a novel bidirectional tree traverse techniques in order to avoid the irrelevant rule for market basket dataset. Second one is to find strong rules from medical dataset using finite automata. Interesting and more informative rules are retrieved from the market basket dataset and medical dataset.

Key words: Data Mining, association rules, Agglomerative Hierarchical clustering, Rule validation, finite automata

1. INTRODUCTION

Data mining is an Artificial Intelligence (AI) powered tool that can discover useful information within a database that can then be used to improve the action [1]. Data mining is an essential step in the process of knowledge discovery in databases in which intelligent methods are applied in order to extract patterns. Many types of "interesting patterns" have been identified in the various research literatures and association rule constitute one such type. Data mining tasks to find these various pattern include characterization, discrimination, association analysis, classification and regression, cluster analysis, outlier analysis and evolution analysis.

Association analysis is the discovery of association rule showing attribute value conditions that occur frequently together in a given set of data. These rules have many applications in areas ranging from e-commerce to sports to census analysis to medical diagnosis. Most of the previous research works [5][6][7][8] are concentrated that the problem of discovering association rules is decomposed into two sub problems. First is finding all frequent itemset in the database, second construct the association rule using frequent item set. Here those kinds of process are produced large number of frequent itemset and association rule [11][12][20][21]. The most of rules are irrelevant to that specified database. Avoid the irrelevant rule is most important when applying the association rule in the database.

We present two different that incorporates to find the valid association rules, and validate them with well-known hierarchical clustering [1][9] [13] to get rules with high accuracy. Our proposed algorithm produces a set of rules that remain valid.

Our new proposed approach goes to the problem of discover the valid association rules from the set of rules derived from frequent itemset mining algorithms [3][4]. This approach addresses to validate the rule. Here we introduce the agglomerative hierarchical clustering technique and tree traversal process for rule validation for market dataset. Normally clustering technique is used to group the data items with similarity in nature. Tree traversal algorithms are used in data structure for memory optimization. But we use these techniques to find the accuracy of the rule. Finite automata are used to validate the token of computer language. But this finite state machine is used to validate the association rules which are derived from medical dataset rules.

This paper is organized as follows. Section 2 Related work of Association Rule mining algorithms, Finite Automata, Clustering and rule validation techniques. Section 3 gives new approaches and its explanation using bidirectional tree traversal for market dataset. The experimental results of rule validation of market dataset are discussed and detailed comparison with the performance analysis of the new approach in Section 4. Section 5 is described about the multidimensional

medical association rules. New Validation procedure is discussed in Section 6 with finite automata. The paper is concluded in the section 7.

2. BASIC PRINCIPLES

2.1 ASSOCIATION RULE MINING

Association rule mining finds interesting association or correlation relationship among a large set of data items with massive amounts of data continuously being collected and stored, many industries are becoming interested in mining association rules from their databases. Let D be a set of n transactions such that D={T_1, T_2, T_3,..,T_n}, where T_i=*I* and I is a set of items, I = (i_1, i_2, i_3, .. ,i_m}. A subset of I containing k items is called a k-itemset. Let X and Y be two itemsets such that X $\subset$ I, Y$\subset$ I, and X $\cap$Y= ϕ. An association rule is an implication denoted by X=>Y where X is called antecedent and Y is called the consequent. We proceed to define association rule metrics. Given an itemset X, support s(X) is defined as the fraction of transactions T_i $\in$D such that X$\subseteq T_i$. Consider P(X) the probability of appearance of X in D, and P(Y|X) the conditional probability of appearance of Y given X. P(X) can estimated as P(X)=s(X). The support of a rule X=>Y is defined as s(X=>Y) = s(XUY). An association rule X=>Y has a measure of reliability called the confidence, defined as c(X=>Y) = s(X=>Y)/s(X). Confidence can be used to estimated P(Y|X): P(Y|X) = P(XUY)/P(X) = c(X=>Y).

2.2. ECLAT ALGORITHM

In Eclat algorithm [16] implementation the set of transactions as a (sparse) bit matrix and intersects rows to determine the support of item sets. The search space of Eclat algorithm is based on depth first traversal of a prefix tree [11].

Éclat principle:-

A convenient way to represent the transactions for the Eclat Algorithm is a bit matrix, in which each row corresponds to an item, each column to a transaction.. A bit is set in this matrix if the item corresponding to the row is contained in the transaction corresponding to the column, otherwise it is cleared. Eclat searches a prefix tree. The transition of a node to its first child consists in constructing a new bit matrix by intersecting the first row with all following rows. For the second child, the second row is intersected with all following rows and so on. The item corresponding to the row is intersected with the following rows to form the common prefix of the item sets, processed in the corresponding child node. Of course, rows corresponding to infrequent item sets should be discarded from the constructed matrix, which can be done most conveniently if it stores with each row the corresponding item identifier rather than relying on an implicit coding of this item identifier in the row index.

2.3. CLUSTERING

Cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters.There are two types of clustering algorithms**: 1. Nonhierarchical and 2. Hierarchical.**

In nonhierarchical clustering, such as the **k-*means*** algorithm, the relationship between clusters is undetermined. Hierarchical clustering repeatedly links pairs of clusters until every data object is included in the hierarchy. With both of these approaches, an important issue is how to determine the similarity between two objects, so that clusters can be formed from objects with a high similarity to each other. Hierarchical clustering creates hierarchy of clusters on the data set. This hierarchical tree shows levels of clustering with each level having a larger number of smaller clusters.

Hierarchical algorithms can be either agglomerative or divisive, that is top-down or bottom-up. All ***agglomerative hierarchical clustering algorithms*** begin with each object as a separate group. ***Hierarchical algorithms*** are rigid in that once a merge has been done, it cannot be undone. Although there are smaller computational costs with this, it can also cause problems if an erroneous merge is done.

Algorithm : Agglomerative hierarchical clustering

Given:
A set *X* of objects {x_1,...,x_n}
A distance function *dis*(c_1,c_2)

1. **for** *i* = 1 to *n*
   c_i = {x_i}
   end for
2. *C* = {c_1,...,c_b}
3. *l* = *n*+1
4. **while** *C*.size > 1 **do**
   a) (c_{min1},c_{min2}) = minimum *dis*(c_i,c_j) for all c_i,c_j in *C*
   b) remove c_{min1} and c_{min2} from *C*
   c) add {c_{min1},c_{min2}} to *C*
   d) *l* = *l* + 1
   end while

2.4. FINITE AUTOMATA

Automata Theory deals with definitions and properties of different types of "computation models". Examples of such models are:

- Finite Automata. These are used in text processing, compilers, hardware design..
- Context-Free Grammars. These are used to define programming languages and in Artificial Intelligence.

3. MARKET BASKET DATASET RULE VALIDATION

Association rules are generated by various algorithms. A large number of rules are generated by these techniques. Some rules are unwanted and uninteresting. Data mining is to mine useful information from very large datasets. Association rules are generated through clustering technique is novel approach for grouping data items. Most of the previous research [24][25] work concentrated only on some statistical approach. In this work we mainly concentrated construction of tree through agglomerative hierarchical clustering. Bidirectional tree traverse is used to find path of the tree. It reduces the execution time.

3.1. ALGORITHMS

Algorithm 3.1: Tree Construction and Assignment of Codes

---

1. Perform Agglomerative Hierarchical Clustering Algorithm for the DataSet DS[...]
   so that items in the data set will be arranged in tree format
2. Enter root node of DS[] . Let r be the current node. Assign code = 0;
3. c<-no of leaves of r. let c be the k value used in clustering, read and assigned at every level
4. nbits <- no of bits required to represent binary value of c
5. for i = 1 to c
   convert binary value of i-1
   for j = 1 to nbits
   do left shift operation on code
   end for
   perform code = code XOR binaryof(i-1) (for node r)
   end for

Second algorithm for rule validation

Algorithm 3.2: Rule Validation

---

1. Decompose the rules to single variables on left and right side
2. Identify Left item on rule from Dataset and assign to L
3. Identify Right item on rule from Dataset and assign to R
4. backtrack L and R towards root. (do parallely)
5. for every traversal of every node from L, increment Lcount
6. for every traversal of every node from R, increment Rcount
7. compare the length of L->Code and R->Code
8. if L->length > R->length
   suspend the operation on L
9. if R->length > L->length
   suspend the operation on R

10. if R = L
    Count <- Lcount + Rcount
11. if Count > support value
    mark the rule as invalid
12 else
    mark the rule as valid

3.2. DESCRIPTION FOR TREE CONSTRUCTION

Associative rule mining, done on data sets would in turn lead to discovery of new rules. Frequency of these rules would vary based on usage. We can not expect uniformity in usage of these rules. Rules need to go for validation in order to ensure its correctness. Some rules may be present less number of times but those rules may contain hidden business intelligence. Some rules may be present much number of times, but they may not be useful in improving the business. In order to identify the rules for improving business, we need to do validation. In this paper, we propose a technique based on hierarchical clustering for validating the rules.
Using agglomerative hierarchical clustering algorithm, items in the dataset are arranged as hierarchical tree. It means items are divided into sub categories at every level. Iteration of this process continues until the leaf nodes contain a single item. Construction of hierarchical tree is called as preprocessing phase. Tree is constructed only once with suitable order so that data items can be accessed easily and logically. Insertion, deletion and modification of data items are also easier as the data structure we have adopted here is "tree".

For example, let us consider the data set contains the following data items.

DS={“pen”,”pencil”,”crayon”,”ink”,”hammer” ,”bread”,”jam”,”milk”,”kids_logos”,”iron_pieces”}

The above data items can be logically arranged as in Figure 1. It can be noted that the items in the data set can only be present at leaf nodes. Other nodes would serve as cluster index. Moreover this phase needs unique number for every node (leaf nodes and non leaf nodes) and binary codes can be used for the purpose. If ‘n’ is the no of child nodes from a node, ‘log n’ no of additional bits will be needed to represent child nodes at next initially. Unique numbers will be assigned from the next level of root node. Initially ‘0’ and ‘1’ can be assigned for the leaves Non Eatables and Eatables. Exploring Eatables, we can assign additional bits ‘00’, ‘01’ and ‘10’ for bread, milk and jam. Concatenating bits at parent node with additional bits, we get ‘100’, ‘101’ and ‘110’ for bread, milk and jam respectively. Similarly we can assign binary code codes for all the nodes and we process the data items as binary digits. Data items (leaf nodes) and cluster indices (non leaf nodes) are converted into binary codes and it can be seen in Figure 1. It can be noted that ‘0’, ‘00’, ‘000’ and ‘0000’ are codes for different items in the tree and they are not treated same. Length of the code is taken for processing phase and not its mathematical value. Moreover every node will contain no of leaf nodes it have, so that it will be useful for later processing. We call this as ‘count’.

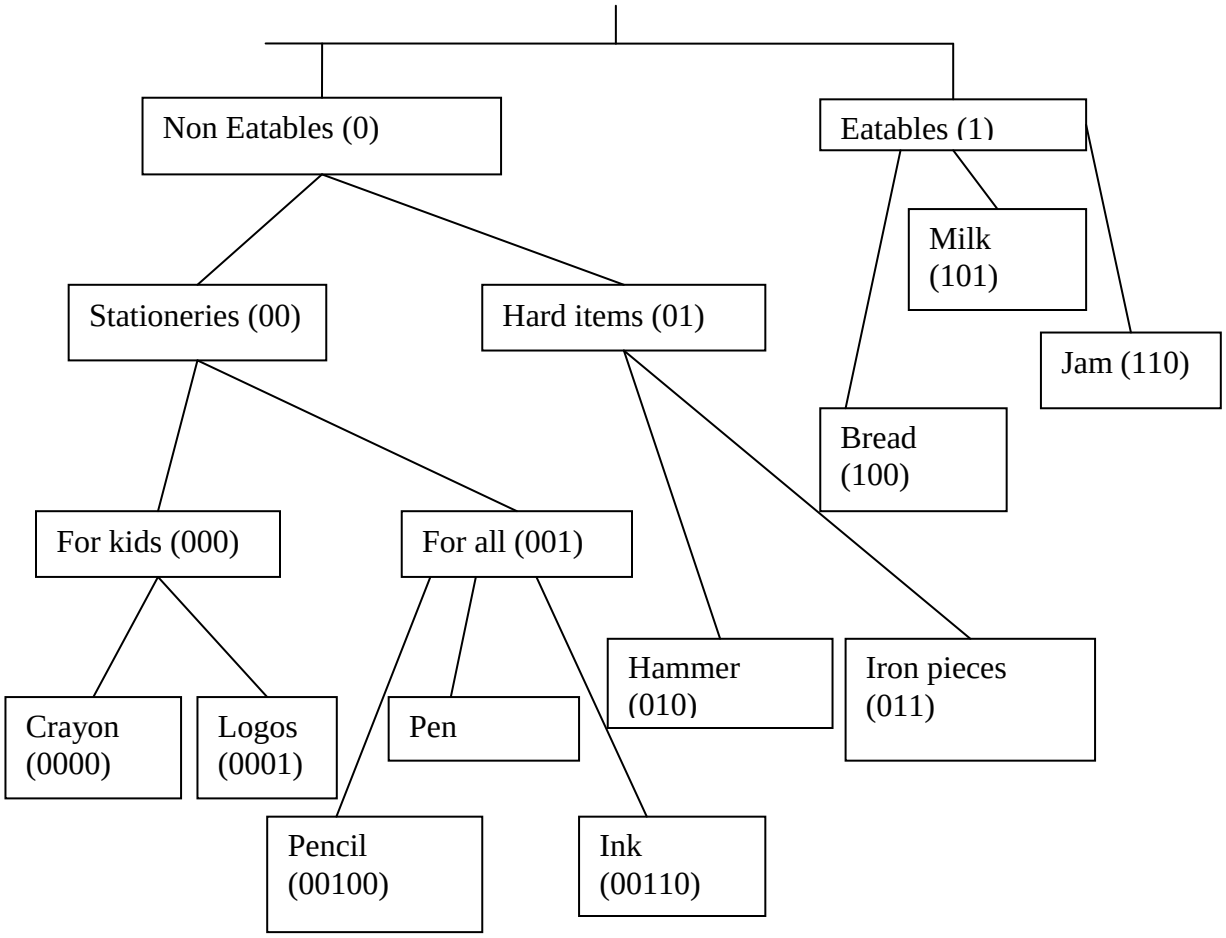

Figure 3.1: Agglomerative Hierarchical clustering for Market data

3.3. DESCRIPTION OF VALIDATION OF RULES:

The outcome of Associated Rule Mining will be described in any of the following form.

1. Pen -> Pencil
2. Pencil -> Pen, Ink
3. Pen -> Pencil, ink

Rule 1 will not raise any conflict for our proposed technique. But rule 2 and rule 3 ought to be decomposed in order to avoid conflicts. Rule 2 can be decomposed into two different rules Pencil -> Pen and Pencil -> Ink. Rule 3 can also be decomposed into Pen -> Pencil and Pen -> Ink. By doing so, we can assure that only one data item will be kept in both L.H.S and R.H.S.

Fix the left item and right item on the tree with help of indices associated with them. Compare the length of the codes on left and right item. If they are equal, simply go back to the parent node. Stepping up to the parent node from left and right is done in parallel. For every step up operation, increment the edge count separately from left and right side. It can be noted that edges counts on left and right items will be zero initially. Once left and right items meet with each other, sum up the edge count and make it as total edge count (TEC). In case the length of the codes on either side won't match with each other, suspend the operation on the side whose code length is smaller. So that bidirectional operational will be stopped temporarily. Compare the length of code on left and right items for all the iterations. If the length matches at any point, suspended bidirectional "step up" operation can be resumed. Once left and right meets with each other, stop the "step up" operation and computer TEC. TEC can be compared with threshold value. If TEC exceeds threshold value, we can mark the processed rule is invalid. Otherwise we can mark it as a valid rule. Threshold value is set based on business strategies. From the above diagram, rule R: crayon -> iron_peices will have TEC value 5. Let us assume based on business strategy threshold value is assigned as 4. In this case our TEC value exceeds the requirement, and the rule R: crayon->iron_pieces can be marked invalid. From simple perception, we can say mostly people wont buy crayon and iron pieces together even though it is recorded once in the history of business transaction. As it won't help in improving business, this transaction can be discarded for framing business strategies.

4. EXPERIMENTAL ANALYSIS

This algorithm applied into the market basket datasets. The training was set at τ = 50 %. Every time the algorithm is run, new samples are created.

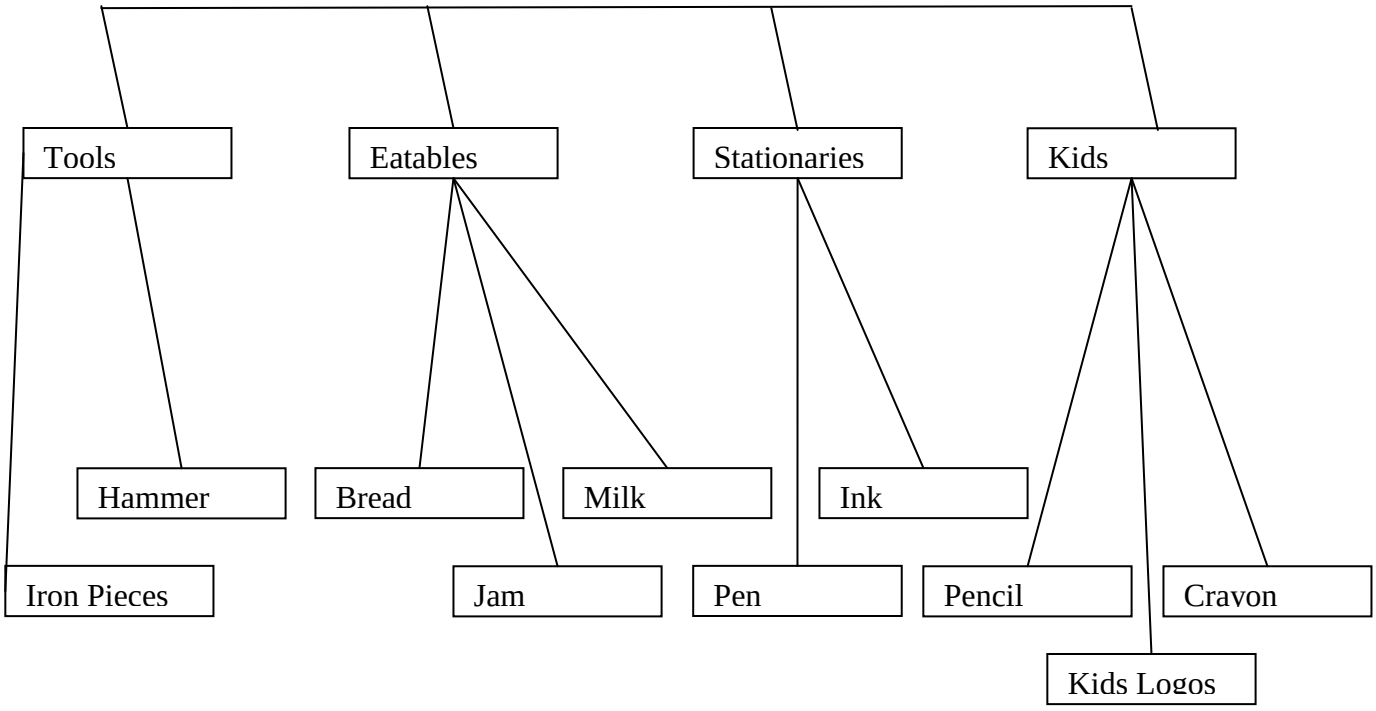

Figure 3.2: shallow representation of tree for sample dataset

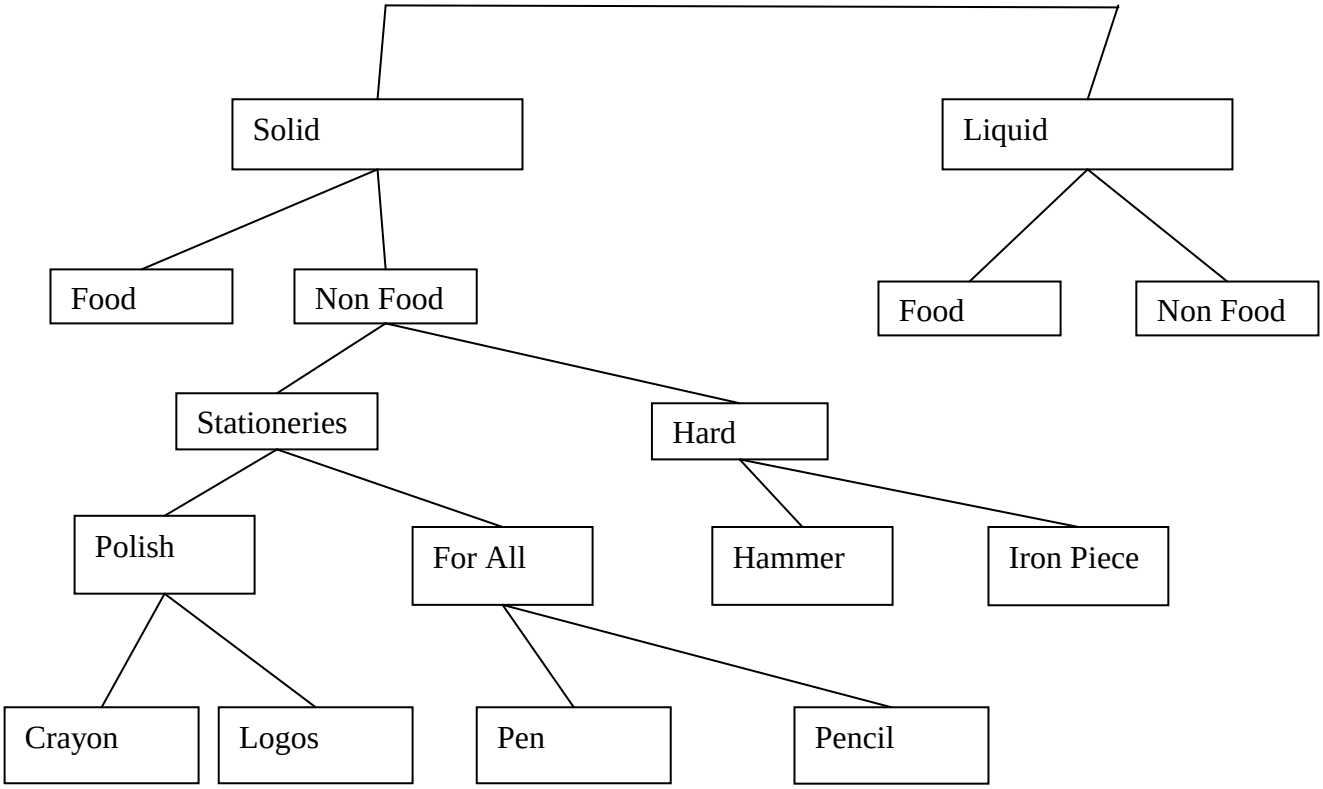

Figure 3: In depth representation of tree for sample dataset

Table 1: Comparison of various types of tree traversal for validation

Sl.No.	Rules	No. of Edges (Fig1)	No. of Edges (Fig2)	No. of Edges (Fig3)
1	Pen → Pencil	2	4	2
2	Pen → cryon	4	4	4
3	Cryon → Iron pieces	5	4	5
4	Milk → Bread	2	2	6
5	Milk → Jam	2	2	6
6	Hammer → Pencil	5	2	5
7	Iron pieces → cryon	5	4	5
8	Ink → Milk	6	4	4
9	Kid Logos → cryon	2	4	2
10	Bread → Iron pieces	5	4	5
11	Ink → Logos	4	4	8
12	Milk → Logos	6	2	8
13	Pen → Ink	2	2	8
14	Hammer → cryon	5	4	5
15	Pencil → ink	2	4	8
16	Logos → iron pieces	5	4	5

In the above table, number of edges for validating the rules has been calculated. Based on threshold value, certain rules can be marked invalid. For example, if the threshold value is set as 5 units, 8 rules will be marked invalid out of 16 rules using figure 1. Hence it can be witnessed that rules are getting validated based on threshold value and construction arrangement of tree. If the threshold value is set as 3 units, 12 rules will be marked invalid out of 16 rules using figure 2.

Figure 2 represents the level of 2 and maximum no. of bits for usage is 4 bits. if the threshold value is set as 6 units, 6 rules will be marked invalid out of 16 rules using figure 3. Figure 3 represents the level is as maximum of 5 and length of bits for usage is 5 bits. From the comparison study table 1 represents total number of edges used in the tree traversal using bi-direction. From the result of figure 1 minimum threshold for generating valid rules is 5 then 8 validated rules are obtained. Figure 2 is a shallow tree. Arrangement in figure 2 holds good for items under individual groups. When a single item overlaps more than one group, shallow tree won't work out properly for rule validation. For example, pencil comes under two different groups, leads pencil to be clustered in a single group and so shallow tree marks the rule pen → pencil as invalid. Even though Figure 3 is a deep tree with different levels; the items are not logically arranged. Hence it marks the rule pen → ink as invalid. Figure 1 logically groups different items and follows necessary exploration at levels marks only false rules as invalid.

Complexity analysis of this algorithm is defined as follows:

Let number of items in dataset be M. let the number of items in transaction be N. Number of bytes required to the present items is log M/2. Space occupied to represent items at each level is 2^n * K. where K is number partition at each level.

Time required to validate a single rule is √log N. as the validation process bi-directional, the complexity is reduced to its square root.

5. MULTIDIMENSIONAL MEDICAL DATASET

The database which we have chosen depicts the complications occurring in diabetes and/or hypertension (increased Blood pressure). All the patients were in the age group of 25 to 70 years, and the sex ratio was almost equal (M:F of 1.1 : 1). All of them had either diabetes or hypertension or both for duration of 10 years and more. They were sub-categorized based on the extent of control of these diseases namely diabetes and hypertension and their complications. The complications are heart disease, kidney disease and stroke. The patient database built has been mined using our approach to predict the diagnosis results.

5.1. RULE ALGORITHMS

This section describes the two types of algorithms in order to construct association rules from medical dataset. First algorithm is used to construct positive rules. The second algorithm deals with to construct both positive and negative rules from frequent and infrequent itemsets.

5.1.1. Positive Rule

This algorithm is constructed by new searching technique, *Bit Search* which scans the transaction only once for k^{th} itemset search. This search technique is used in eclat algorithm in finding both positive and negative association rules. This algorithm shows usage of sparse matrix to mine association rules with bit array data structure. This produces only positive association rules.

5.1.2. Negative Rule

From the algorithm 1, Positive rules are only constructed which yields only small number of rules. This will yield only a small amount of useful information is mined from the dataset. Algorithm 2 describes about how to construct the negative rules with the help of eclat procedure with new searching technique. This BitMaskNegativePos mines most useful associations among itemsets. In general, medical diagnosis system is complicated one. So that construction of more rules with higher confidence is a major task of this work.

Algorithm 1: BitMaskPos

1. Initialize the matrix bit[n][m] where *n*-> number of itemsets *m*-> no. of transactions
2. For each item in the transaction repeat the steps 3,4,5
3. for each transaction in the input file repeat the step 4
4. Check whether the bit[item][m] is not equal to zero. If yes the increment the total count (tcount)
5. Calculate the support using total count divided by no. of transactions
6. for each transaction in Bit array repeat the step 2

7. AND can be used to find the result value for subset with transaction dataset. If the result value is as same as the subset value, the *k*-itemsets are present in the transaction
8. increment Bit_itemset count
9. check whether bit_item_count is greater than or equal to minsup. If yes add the frequent bit_itemsets otherwise delete item
10. calculate confidence measure using count divided by tcount
11. Find Positive rules by using minimum threshold for support value

Algorithm 2: BitMaskNegativePos

1. Initialize bit[n][m] where *n*-> number of itemsets in frequent datasets *m*-> no.of transactions and Initialize *nbit*[*n1*][*m*] where n1 -> number of itemsets in infrequent datasets
2. for each item in itemsets Repeat 3
3. for every non zero entry in the matrix repeat step 4,5
4. Increment the tcount
5. Check whether the frequent item matches with frequent item. If yes, increment the count
6. Calculate support measure for positive rule
7. Calculate confidence measure positive rule
8. For each item in itemsets repeat step 9,12,13
9. For every non-zero entry in the infrequent item matrix, repeat step 10,11
10. Increment the tcount
11. Check whether two infrequent item matches or infrequent matches with a frequent item. If yes increment count
12. Calculate support measure for negative rules
13. Calculate confidence measure for negative rules

5.2. EXPERIMENT AND EVALUATION

These algorithms are implemented in the medical dataset which are described in the section 2. The table 5.1 shows the sample medical data sets which the patients are affected by the diseases Hypertension and diabetes with their complications. Both the algorithms implemented in this dataset and produce association rules. In the table 1, DM indicates Diabete disease, HT indicates Hypertension and dz means disease.

Table 5.1: Sample Medical Transaction data

Transaction id	Item set	Numeric data set
T1	DM,HT,heartdz	1,2,3
T2	DM,Heartdz,kidneydz	1,3,4
T3	HT,Heartdz,kidneydz	2,3,4
T4	DM,kidneydz	1,4
T5	HT,heartdz	2,3
T6	HT,stroke	2,5
T7	HT,heartdz,stroke	2,3,5
T8	DM,heartdz	1,3
T9	HT,kidneydz	2,4
T10	DM,HT,kidneydz	1,2,4

5.3. CONCEPT HIERARCHY OF MEDICAL DATASET

A concept hierarchy defines a sequence of mappings from a set of low level concepts to higher level more general concepts. Concept hierarchies may also be defined by grouping values for a given dimension or attribute resulting in a set grouping hierarchy. Concept hierarchies allow data to be handled at varying levels of abstraction. A top-down progressive deepening method is developed for efficient mining of multiple-level association rules from large transaction databases.

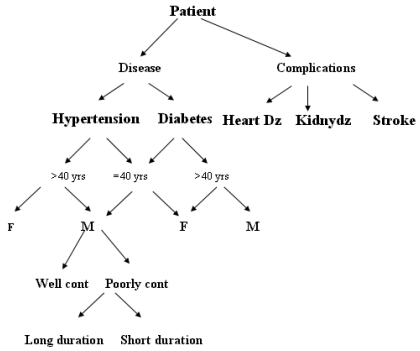

Figure 5.1 Concept hierarchy of Medical Database

Figure 5.1 represents the database in a hierarchical tree structured approach. A patient who is affected by either one of the diseases has a certain chance of developing complications. Using concept hierarchy, diseases are classified with respect to age, sex, duration of disease and the extent of control of the disease. Each has its own minimum support at each level.

Figure 5.1 represent the new approach of Medical Database.

The problem is focused on mining a patient database using the following parameters:

i)Age ii) Gender iii) Level of Control iv)Duration of disease

The above parameters are used for mining the patient history for the case of disease due to hypertension, diabetes and complications including kidney dz, heart dz, and stroke. The problem scope is focused and limited to 48 combinations. All these combination data items are infrequent in nature. Infrequent itemsets generate only negative rules.

Table 5.2: Bit conversion of Medical Dataset

Factors				**Diseases**		**Complications**		
Gender	**Age > 40 yrs**	**Duration**	**Level of control**	**Hyper-tension**	**Diabetes**	**Kidney disease**	**Heart disease**	**Stroke**
1	1	1	0	1	0	0	1	0
0	**1**	**0**	**1**	**1**	**1**	**1**	**1**	**1**
1	0	0	1	0	1	1	0	0

Table 5.3: Infrequent itemset combination

Bit Value	**Itemset value**	**Bit Value**	**Itemset value**
000001	1	100001	25
000010	2	100010	26
000011	3	100011	27
000101	4	100101	28
000110	5	100110	29
000111	6	100111	30
001001	7	101001	31
001010	8	101010	32
001011	9	101011	33
001101	10	101101	34
001110	11	101110	35
001111	12	101111	36
010001	13	110001	37
010010	14	110010	38
010011	15	110011	39
010101	16	110101	40
010110	17	110110	41
010111	18	110111	42
011001	19	111001	43
011010	20	111010	44
011011	21	111011	45
011101	22	111101	46
011110	23	111110	47
011111	24	111111	48

Table 5.2 represents the conversion of patient database into bit table. In the table for gender 1 represents Male and 0 as Female, Age > 40 as for others 0. Duration is classified into Long duration which is more than 10 years represented as 1 and short duration as 0. Patient control over the disease is converted into two types are Poorly control as 1 and Well control as 0.

The diseases and factors are combined and construct the new dataset. The table 4 shows the infrequent itemset bit combination of new data items. With the help of Bit value representation, the new medical data set is generated. First four digit of bit sequence represents the factors and last two bit as diseases hypertension and diabetes. All these six bits are combined with various

combinations and assign as new item. Complications are separated by individual items.

From he table 4 the factors and diseases are combined as the example as follows.
Long duration male above 40 yrs poorly controlled hypertension diabetes having complications

For any patient, the diseases which are taken affected by the above factors will affect the health of the patient with severe complications. The multi dimensional dataset creation will yield useful rules which are more informative. More infrequent items are created with only two disease and four factors.

6. FINITE AUTOMATA RULE VALIDATION

This section presents a novel approach for knowledge discovery from sequential data. Instead of mining the examples in their sequential form, we suppose they have been processed by a machine learning algorithm that has generalized them into a deterministic finite automaton (DFA). Thus, we present a theoretical framework to extract decision rules from this DFA.

6.1. FINITE STATE MACHINE\

A **finite-state automaton or finite state machine (FSM)** is a mathematical abstraction sometimes used to design digital logic or computer programs. It is a behavior model composed of a finite number of states, transitions between those states, and actions, similar to a flow graph in which one can inspect the way logic runs when certain conditions are met. It has finite internal memory, an input feature that reads symbols in a sequence, one at a time without going backward; and an output feature, which may be in the form of a user interface, once the model is implemented. The operation of an FSM begins from one of the states is called a *start state*, goes through transitions depending on input to different states and can end in any of those available, however only a certain set of states mark a successful flow of operation is called *accept states*.

A current *state* is determined by past states of the system. As such, it can be said to record information about the past, i.e., it reflects the input changes from the system start to the present moment. The number and names of the states typically depend on the different possible states. A *transition* indicates a state change and is described by a condition that would need to be fulfilled to enable the transition. An *action* is a description of an activity that is to be performed at a given moment. There are several action types:

Entry action
: which is performed *when entering* the state

Exit action
: which is performed *when exiting* the state

Input action
: which is performed depending on present state and input conditions

Transition action which is performed when performing a certain transition

Acceptors and **recognizers** (also **sequence detectors**) produce a binary output, saying either *yes* or *no* to answer whether the input is accepted by the machine or not. All states of the FSM are said to be either accepting or not accepting. At the time when all input is processed, if the current state is an accepting state, the input is accepted; otherwise it is rejected. As a rule the input are symbols (characters); actions are not used. The figure 6.1 shows a finite state machine which accepts the Multidimensional medical association rules. The rules which are passes through the FSM are strong and informative so as to validate the rules. If the rules are not passed though the FSM is not a valid one.

6.2. MULTIDIMENSIONAL ASSOCIATION RULES

With the help of Multidimensional approach and concept hierarchy process actually creates sets of group of data. For mining multidimensional association rules, multiple minimum support should be provided for generalizing rules. This may lead to the generation of many low confidence associations such as "**hypertension➔heart disease**" and the discovery of some interesting ones, such as **">40 yrs old Long duration Poorly controlled male hypertension ➔heart disease" with low support and higher confidence**.

For this multidimensional approach all the above 48 combinations are infrequent items. So the infrequent items generate only negative association rules which yield interesting information. 100% confidence provides intelligent information for crucial decisions. 75% to 99% confidence yields better informative rules. Lower confidence which is less than 75% gives no so much information. BitMaskPos and BitMaskNegativePos are algorithms to generate both Positive and Negative association rules respectively. The two types of datasets are implemented with the help of the above mentioned algorithms are shown in figures.

Figure 6.1 to figure 6.3 represent various rules comparison of multidimensional dataset with the existing single dimension rules. All the figures are divided by percentage of confidence and numbers of rules are generated. Confidence percentage is divided as 100%, 75% to 99% and less than 75%. The medical dataset is implemented in both types, that is the data as it is and converted new multidimensional data set. The rules are generated in several combinations of itemsets such as 1) Infrequent, frequent, frequent and 2)Infrequent, frequent.

Figure 6.1 shows the Rules which are having Infrequent, Frequent and Frequent itemsets rules between multidimensional and single dimensional medical datasets. From this figure, a large number of rules are generated with higher confidence as 100%. Nearly 60% of the multidimensional rules are having the higher confidence 100% while comparing with single dimensional dataset having 5% of rules only. So, the multidimensional data set produces the highly informative negative association rules. For any medical database each and every combination of dataset is essential to protect human being. Hence, more hidden information are retrieved from the medical database.

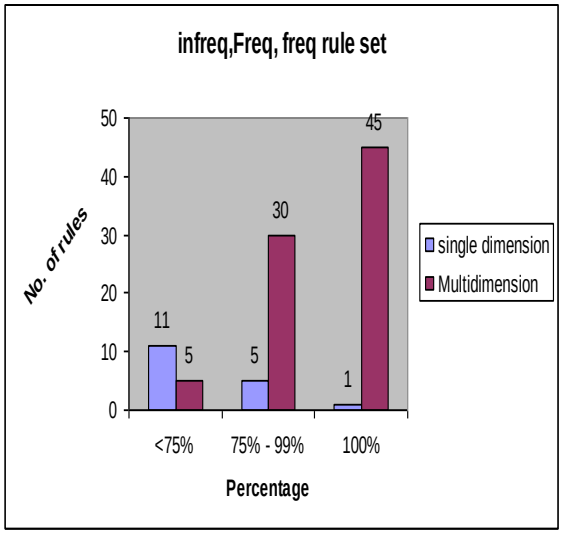

Figure 6.1: Infrequent, Frequent, Frequent itemsets rules

Figure 6.2 represents the infrequent and frequent itemsets rules and their comparisons among confidence measurements. There are no single dimensional rules from this implementation. 60% of the rules are having higher confidence of 100% and 75% to 99%. Hence, intelligent rules are generated from this infrequent combination. There are 57 rules are produced with the help of this combination.

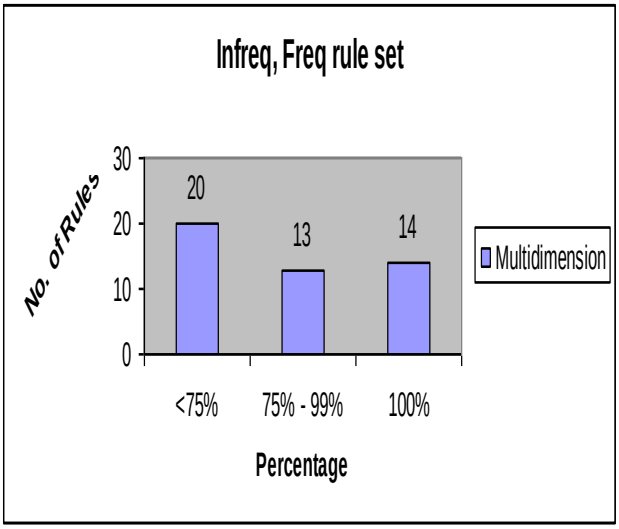

Figure 6.2: Infrequent, Frequent itemsets rules

From the figures 6.2 & figure 6.3, infrequent itemsets combinations produces negative associations only generates more number of rules. Hence, multidimensional datasets only supports the patients information in highly informative manner in both positive and negative associations.

Figure 6.3 represents the rules produced by multidimensional data sets in both positive and negative combinations rule measurements. From the diagrammatical representation, a large number of negative rules are observed with higher confidence as 100%. This will yield more useful information for further investigation. From this diagram, infrequent item combination yields large number of rules while compare with others. The rules obtained from the multidimensional dataset are shown in the figure. Among all types of rules, highly informative pattern rules are obtained through infrequent items combination with frequent items which are negative in nature.

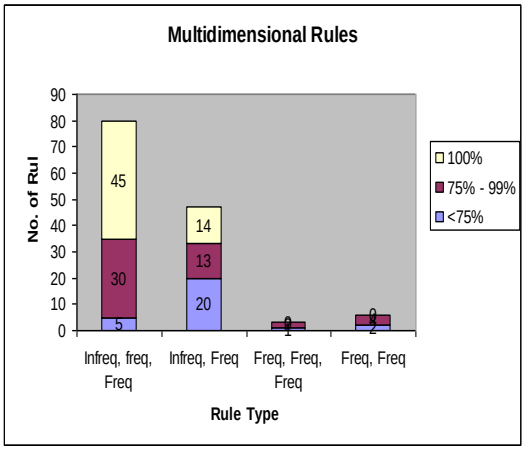

Figure 6.3: Multidimensional rules

This observation lead us to examine the methods for mining association rules at multidimensional dataset with minimum support using infrequent itemset, which may only discover highly pattern rules at different levels but also have high potential to find nontrivial informative association rules because of its flexibility at focusing the attention to different sets of data.
Even though all the rules predicts highly informative, some rules are not in the proper form of disease and Finite automata construction. The following subsection covers the valiadation of the associations among items.

6.3. MEDICAL DATASET FINITE STATE MACHINE

The validation plays the vital role to retrieve highly informative strong rules from the medical dataset. Finite state machine with the help of medical indication and report from the experts extracts only strong association rules. In the figure 6.4, 'o' represents the starting state of the Finite automata. Nodes 1 to 6 are called as intermediate nodes. 'A 'represents patient with no complication (NC). Let B indicates he complication of Heart disease (Hdz). Let C be represented as Heart disease and Kidney disease (Hdz + Kiddz). D is treated as Heart disease, Kidney disease and Stroke (Hdz + Kid dz + St). Let E is represented as Heart disease and stroke. Hdz+St (A to E – Final States), Let I – Poorly controlled diabetic PC+D,, j – poorly controlled hypertension (PC+HT), k – Poorly controlled diabete and hypertension (PC+DHT) and l – Well controlled Diabete are assigned for easy computation process for this finite automation. The following representations are assigned the valid tokens of the string for easy manipulation of the dataset: m- Well controlled Hypertension (m – WC+HT), n- well controlled diabete and hypertension (n – WC+DHT) , o indicates the default, p indicates the – else part, q - Age>40 && Duration >10, r – not (Age<40 && Dur<10) and s – Age<40 && Duration>10.

6.4. VALIDATED ASSOCIATION RULES

Rule **hypertension → heart disease** as single dimension will yield so many rules in multidimensional

1. *.Long duration male above 40 yrs poorly controlled hypertension diabetes (heart disease*
2. *.short duration male above 40 yrs poorly controlled hypertension diabetes (heart disease*
3 *.Long duration female above 40 yrs poorly controlled hypertension diabetes (heart disease,*
4. *.Long duration male below 40 yrs poorly controlled hypertension diabetes (heart disease*
5. *Long duration male above 40 yrs Well controlled hypertension, stroke (heart disease*
6. *Long duration male above 40 yrs Poorly controlled diabetes, stroke (heart disease*
7. *Long duration male above 40 yrs Poorly controlled diabetes, stroke (kidney disease, heart disease*
8. *Long duration female above 40 yrs poorly controlled diabetes and hypertension (heart disease*

The above rules are validated which are derived with the help of Medical dataset Finite state machine.

0 – Start state, 1 to 6 – Intermediate States, A – NC, B – Hdz, C – Hdz+Kdz,
D – Hdz+Kdz+St,
E – Hdz+St (A to E – Final States), I – PC+D, j – PC+HT, k – PC+DHT, l – WC+D,
m – WC+HT,
n – WC+DHT, o – default, p – else, q - Age>40 && Duration >10,
r – not (Age<40 && Dur<10),
s – Age<40 && Duration>10

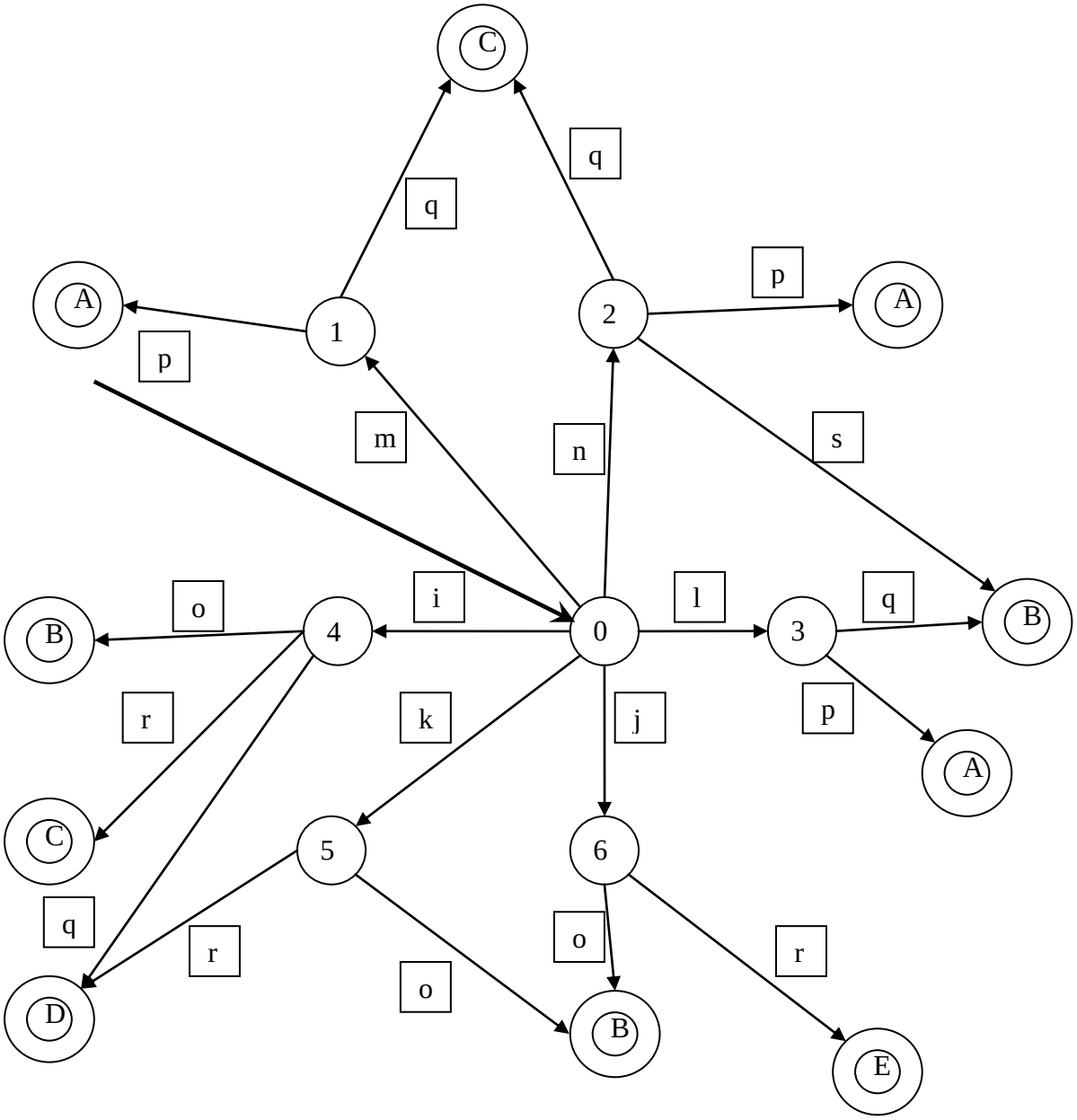

Figure 6.4: Finite State Machine for Medical Dataset

7. CONCLUSION

This paper focused on two main research issues. The first issue is to construct tree structure using hierarchical clustering technique with the help of association rule for the market basket datasets. The second issue is the validation of medical rules using finite state machine on an independent set, which is required to eliminate unreliable rules from health care dataset. Experiments on a real data set studied the impact of constraints and elimination of unreliable rules with validation with minimum threshold value. So, from these practical examples, we conclude that construction of tree and finite state automata plays the major roles. These methods may provide more efficiency compare to other process.

REFERENCES

1 Mannila, H., Toivonen, H., and Verkamo, A. I. 1995. Discovering frequent episodes I in sequences. In *Interantional Conference on Knowledge Discovery and Data Mining*. IEEE Computer Society Press.

2 Agrawal, R., Imielinski, T., and Swami, A. N. 1993. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, P. Buneman and S. Jajodia, Eds. Washington, D.C.,

3 Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, J. B. Bocca, M. Jarke, and C. Zaniolo, Eds. Morgan Kaufmann, 487-499.

4 Agrawal, R. and Srikant, R. 1995. Mining sequential patterns. In *Eleventh International Conference on Data Engineering*, P. S. Yu and A. S. P. Chen, Eds. IEEE Computer Society Press, Taipei, Taiwan, 3-14.

5 Bayardo, R., Agrawal, R., and Gunopulos, D. 1999. Constraint-based rule mining in large,dense databases. Berkhin, P. 2002. Survey of clustering data mining techniques. Tech. rep., Accrue Software, San Jose, CA.

6 Cheung, D. W.-L., Lee, S. D., and Kao, B. 1997. A general incremental technique for maintaining discovered association rules. In *Database Systems for Advanced Applications*. 185-194.

7 Das, A., Ng, W.-K., and Woon, Y.-K. 2001. Rapid association rule mining. In *Proceedings of the tenth international conference on Information and knowledge management*. ACM Press, 474-481.

8 Han, J. and Fu, Y. 1995. Discovery of multiple-level association rules from large databases. In *Proc. of 1995 Int'l Conf. on Very Large Data Bases (VLDB'95), ZÄurich, Switzerland, September 1995*. 420-431.

9 Han, J. and Kamber, M. 2000. *Data Mining Concepts and Techniques*. Morgan Kanufmann. Han, J., Koperski, K., and Stefanovic, N. 1997. GeoMiner: a system prototype for spatial data mining. 553-556.

10 Han, J., Pei, J., and Yin, Y. 2000. Mining frequent patterns without candidate generation. In *2000 ACM SIGMOD Intl. Conference on Management of Data*, W. Chen, J. Naughton, and P. A. Bernstein, Eds. ACM Press, 1{12. James, M. 1985. *Classification Algorithms*. Wiley&Sons, Inc.

11 Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A. I. 1994. Finding interesting rules from large sets of discovered association rules. In *Third International Conference on Information and Knowledge Management (CIKM'94)*, N. R. Adam, B. K. Bhargava, and Y. Yesha, Eds. ACM Press, 401-407.

12 Lee, S. D. and Cheung, D. W.-L. 1997. Maintenance of discovered association rules: When to update? In *Research Issues on Data Mining and Knowledge Discovery*. 0-14.

13 Lent, B., Swami, A. N., and Widom, J. 1997. Clustering association rules. In *ICDE*. 220{231. Lu, H., Han, J., and Feng, L. 1998. Stock movement prediction and n-dimensional intertransaction association rules.

14 Zhi-Choa Li, Pi-Lian He, Ming Lei, "A High Efficient AprioriTID Algorithm for mining Association rule", Proceedings of 4th International Conference on machine learning and cybernetics,18-21 AUG 2005.

15 Ng, R. T., Lakshmanan, L. V. S., Han, J., and Pang, A. 1998. Exploratory mining and pruning optimizations of constrained associations rules. 13-24.

16 Psaila, G. and Lanzi, P. L. 2000. Hierarchy-based mining of association rules in data warehouses. In *Proceedings of the 2000 ACM symposium on Applied computing 2000*. ACM Press, 307-312.

17 Savesere, A., Omiecinski, E., and Navathe, S. 1995. An efficient algorithm for mining association rules in large databases. In *Proceedings of 20th International Conference on VLDB*.

18 Smythe and Goodman. 1992. An information theoretic approach to rule induction from databases. In *IEEE Transactions on Knowledge and Data Engineering*. IEEE Computer Society Press.

19 Srikant, R., Vu, Q., and Agrawal, R. 1997. Mining association rules with item constraints. In *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining, KDD*, D. Heckerman, H. Mannila,

20 Mingju Song and Sanguthevar Rajasekaran *A transaction mapping for frequent itemsets mining* IEEE transactions on Knowledge and Data Engineering 18(4):472-480, April 2006.

21 Ke Su, Fengsdhan Bai *Mining weighted Association Rules* IEEE transactions on KDE 489-5=495, April 2008

22 Balaji Padmanabhan, Alexandar Tuzhilin *On Characterization and Discovery of Minimal unexpected pattern in Rule Discovery* IEEE trans on KDE vol 18 no. 2 Feb '06

23 Rajeev Rastogi, Kyuseok Shim *Mining Optimized Association Rules with categorical and numerical attributes* IEEE trans on KDE vol 14 No.1 Jan/Feb '02

24 Dr. E. Ramaraj, et al **Strong Rule Mining algorithm with validation** published at International Conference on SOFTECH07, Avinasilingam University, Coimbatore, dated 24-25th Jan 2007.

25 Stephane Lallich et al, *Association Rule interestingness: measure and statistical validation*

26 Zhou X, Chen S, Liu B, Zhang R, Wang Y, Li P, Guo Y, Zhang H, Gao Z, Yan X. 'Development of traditional Chinese medicine clinical data warehouse for medical knowledge discovery and decision support' Artificial Intelligent Med. 2010 Feb-Mar;48(2-3):139-52.

27 Michael sisper, Introduction to theory of Computation