

An Inference Mechanism Framework for Association Rule Mining

Kapil Chaturvedi

Department of Computer Application
Rajiv Gandhi Proudयोगiki
Vishwavidyalaya
Bhopal, MP, India

Dr. Ravindra Patel

Department of Computer Application
Rajiv Gandhi Proudयोगiki
Vishwavidyalaya, Bhopal, MP, India

Dr. D.K. Swami

Faculty of Engineering
VNS Group of Institutions
Bhopal, MP, India

Abstract—Available approaches for Association Rule Mining (ARM) generates a large number of association rules, these rules may be trivial and redundant and also such rules are difficult to manage and understand for the users. If we consider their complexity, then it consumes lots of time and memory. Sometimes decision making is impossible for such kinds of association rules. An inference approach is required to resolve this kind of problem and to produce an interesting knowledge for the user. In this paper, we present an inference mechanism framework for ARM, which would be capable enough for resolving such problems, it would also predict future possibilities using Markov predictor by analyzing available fact and inference rules.

Keywords—Inference rules; ARM; Knowledgebase; Expert System

I. INTRODUCTION

Association rule mining (ARM) is the well-researched data mining technique [7, 9]. Most popular ARM application are market basket analysis, which uses a rule based knowledge representation which refer to the relationship between objects, it was first introduced in 1993 [2], in 1994 R. Agrawal and R. Srikant provided a candidate generation based technique formally Apriori algorithm [1] to generate rules, it outperforms when support count is high and number of items are less. The second approach for ARM is Frequent Pattern growth mining formally FP- Growth approach [10] proposed by J. Han, J. Pei and Y.

In 2000, it is two pass technique where in first pass it counts the number of occurrences of objects and second pass generates the Frequent Pattern tree (FP-tree), FP-Growth outperform when support count is low, but it requires much storage to design and store a tree structure space in case large transaction set is given. Other approaches are matrix based approaches which use Boolean logical and arithmetic operations to generate association rules [18, 11, 23, 8, 19, 16, 4, 24] the pros of Boolean matrix based approaches are - It consumes less memory due to their bit data format and makes possible to access and process the huge Boolean relational database to generate frequent patterns. ARM algorithm uses interesting measures like support, confidence and additional measures are Lift and Conviction.

The major problem with association rule mining approach is that, it generates a huge number of rules that may be redundant and insignificant; here the decision making process is complex due to these useless rules so there is a need of an

approach which is capable to find interesting rules to take inference decision. In this paper, we propose an inference mechanism framework for association rule mining, which analyzes the association rules and generate inference rules as well as future possibilities [5] using the Markov predictor.

The rest of this paper is organized as follows: Section 2 discusses the discovery of strong association rules, section 3 gives a brief overview of the inference mechanism in rule based systems, section 4 discusses related work and literature review, section 5 presents a detailed description of the proposed inference framework for ARM, section 6 explains problem with a real time example of medical database, section 7 discusses about obtaining results and section 8 finally concluded the paper.

II. DISCOVERY OF STRONG ASSOCIATION RULES

Definition 1: Let I is a set of items which contains different items $I_1, I_2, I_3, \dots, I_n$ which may occur in different transactions, $I = \{I_1, I_2, I_3, \dots, I_n\}$.

Definition 2: Let T is set of transactions contains different transactions $t_1, t_2, t_3, \dots, t_m$: $T = \{t_1, t_2, t_3, \dots, t_m\}$ where $T \subseteq I$ in transactional data base D .

Definition 3: An association rule represented in the form of implication of $X \rightarrow Y$ where $X, Y \subset I, X \cap Y = \emptyset, I$ is set of items, X is called the antecedent and Y is called consequent.

Definition 4: Let S is the support and C is confidence, then $X \rightarrow Y$ is said to be an association rule, if the minimum support count $S(X \rightarrow Y) \geq \text{Min}(S)$ and minimum confidence $C(X \rightarrow Y) \geq \text{Min}(C)$.

Definition 5: Support (S) and Confidence (C) are two important measures of Association rule mining for finding interesting and useful items from user concern, user predefines the thresholds (Minimum support and Minimum confidence) to drop un-useful and uninteresting rules.

Definition 6: Support (S) of an association rule is defined as the percentage of records that contain $X \cup Y$ to the total number of records in the database. Suppose the support of an item is 20%, it means only 20 percent of the transaction contains purchasing of this item.

Support is the probability of occurrence of $X \cup Y$ in the total number of transactions

$$\text{Support, } S(X \rightarrow Y) = \text{Prob}(X \cup Y)$$

Definition 7: Confidence of an association rule is defined as the percentage of the number of transactions that contain $X \cup Y$ to the total number of records that contain X.

Confidence is a measure of strength of the association rules, suppose the confidence of the association rule $X \rightarrow Y$ is 80%, it means that 80% of the transactions that contain X also contains Y together.

Confidence,

$$C(X \rightarrow Y) = [\text{Prob}(X \cup Y) / \text{Prob}(X)]$$

To find frequent patterns and discover interesting rule also uses some additional measure like Lift and conviction.

Definition 8: Lift is defined as “ratio of the observed support to that expected (if A & B were independent)”

$$\text{Lift}(X \rightarrow Y) = \text{Prob}(X \cup Y) / \text{Prob}(X) \times \text{Prob}(Y)$$

$\text{Lift}(X \rightarrow Y) > 1$: So that X and Y are positively correlated, i.e. the occurrence of one implies the occurrence of the other.

$\text{Lift}(X \rightarrow Y) < 1$: So that the occurrence of X is negatively correlated (or discourages) with the occurrence of Y.

$\text{Lift}(X \rightarrow Y) = 1$: So that X and Y are independent and there is no correlation between them.

Definition 9: Conviction is the ratio of the expected frequency of occurrence of X without Y, that means “the frequency that the rule makes an incorrect prediction (if X & Y were independent)”

$$\text{Conviction}(X \rightarrow Y) = 1 - \text{Supp}(X) / [1 - \text{Conf}(X \rightarrow Y)]$$

Properties of a good measure.

- 1) $P(X \wedge Y) = P(X) \times P(Y)$ – Statistically Independent
- 2) $P(X \wedge Y) > P(X) \times P(Y)$ – Statistical Correlated
- 3) $P(X \wedge Y) < P(X) \times P(Y)$ – Negatively Correlated

III. INFERENCE MECHANISM IN RULE BASED SYSTEMS

A. Inference Mechanism

In the branch of knowledge engineering and artificial intelligence an inference mechanism is an approach that helps to drive answer from the knowledge base [6]; An inference mechanism works as a control strategy in decision making system, it processes the knowledge base by applying given facts to derive new knowledge, it uses reasoning by matching and unification of similarity between the objects. An inference rule has two parts, an “IF” closure and a “THEN” closure, for example, if a patient has symptoms S1, S2, S3 then he/she has the Disease1

$$\text{i.e. } X(S1) \wedge X(S2) \wedge X(S3) \rightarrow X(\text{Disease1})$$

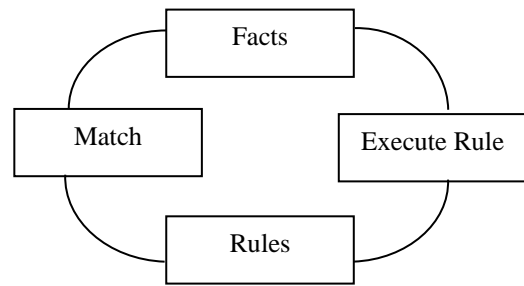


Fig. 1. Traditional inference approach

- **Forward Chaining** - It compares each rule stored in the knowledge base with the given facts stored in the database. When the IF part or antecedents of the rule matches the fact then the rule is fixed and its fact part is executed.

e.g.

- Rule -1: If D and E Then F
- Rule -2: If A and B and C Then D

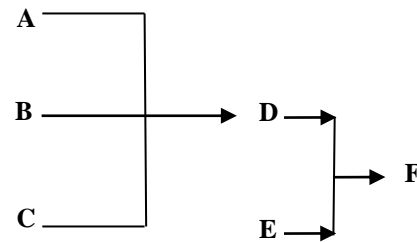


Fig. 2. Search strategy of forward chaining system

Data driven reasoning

- 1) Start with known data (fact).
- 2) Fires the rule that has an antecedent that matches the facts in the database and add any reasoning facts to the database

Each rule can fire only once.

When no more rules can fire, then stop.

- **Backward Chaining** – The inference engine works backward from a conclusion to be proven to determine if there is data in the workspace to prove the truth of the conclusion.

e.g.

- Rule - 1: If D and E Then C
- Rule - 2: If D Then B
- Rule - 3: If B and C Then A

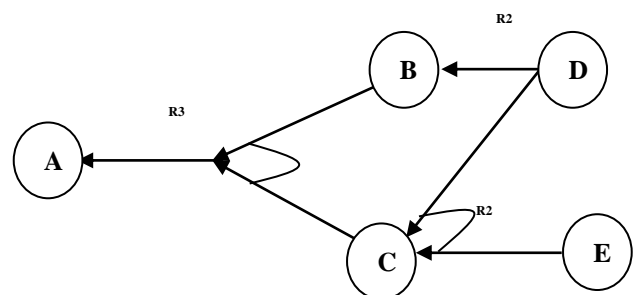
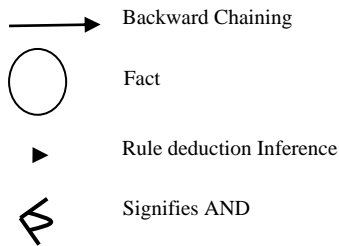


Fig. 3. Search strategy of backward chaining system



IV. LITERATURE REVIEW

Many approaches are well investigated in the literature for Association Rule Mining, these approaches are based on candidate generation [1], tree based approaches [10] and matrix based approaches [8, 11, 19, 24]. But the limitation with ARM algorithms is that it produces a huge number of rules that might be superfluous, dead and also useless to overcome this problem some innovative approaches are well investigated in the literature as a rule based inference mechanisms these are as follows.

Chang-Hung Lee et al. proposed an algorithm PPM (Progressive-Partition-Miner) for mining general temporal association rules in publication databases [17], in this they first partitioned the publication database on the basis of exhibition periods of items, in first scan it produces candidate 2-itemset which are used to generate K-temporal item sets and K-sub items set, in second scan it generates frequent K-temporal item sets and K-sub items set this algorithm employed scan reduction technique to effectively reduce the number of database scans.

In [22] Jian-Bo Yang et al. proposed an approach generic rule-based inference methodology using the evidential reasoning (RIMER) in this they proposed a new knowledge representation scheme in a rule base by analyzing existing knowledge base structure using a belief structure. In [21] S. Venus proposed a rule based backward chaining inference engine which is an Arabic expert system based approach on natural language for diagnosing diseases.

Some built in rule based inference tools are as follows:

A. JESS (Java Expert System Shell) –

Jess is a Rule based inference engine which developed in a scripting language environment written in Oracle's Java language by Ernest Friedman-Hill [14] at Sandia National Laboratories in Livermore, CA, it uses a rule based reasoning algorithm to find inference, user can use it by just adding Jess package in java library and can use its feature by adding java's APIs in there java implementation, it is a light weighted and faster rule based engine.

B. Apache Jena –

Jena is an open source semantic web framework [12] for java, it provides inference java API to use in configuring own inference rules, it facilitate to work with models like RDFS (Resource Description Framework) and OWL (Web Ontology Language) to add extra semantics to users resource description framework data.

C. BaseVisor –

BaseVisor is a closed source rule based forward Chaining inference engine[15], it handles fact in the form of resource description framework (RDF) that triples with support for web ontology language(OWL) and XML schema data types, BaseVisor provides java API to add new features. It requires JRE 1.5. BaseVisor2.0 has following features OWL 2 RL processing, rule and query support, user defined function, user friendly syntax, inclusion mechanism.

D. SweetRules –

Semantic web forward Chaining is an open source inference engine[13] which uses rule based reasoning algorithm for SWRL and ontology. It has reason of SWRL(semantic web rules Language) and RuleML (Rule Markup/Modeling Language) and It is a tool for reasoning

E. OWLIM –

OWLIM is the most efficient semantic repository [13] or a robust inference engine implemented in java with advanced features which is able to load huge number of Resource Description Framework (RDF) statements, it is packaged as SAIL (storage and inference layer), available in two additions BigOWLIM, SwiftOWLIM(free to download and use). Basically it is a RDF database management system which has high scalability, loading and query evaluation performance so it is used in research projects and software tools.

V. PROPOSED FRAMEWORK FOR INFERENCE MECHANISM

In this approach we are proposing an association rule based inference mechanism framework, it works in five phases as shown in figure-4.

A. Data pre-processing & Features extraction –

This model first preprocess the dataset to map data in the required format by mapping objects/items with appropriate index values for further smooth processing and then examine the existing dataset and extract the features of dataset to decide which ARM approach/algorithm would be most suitable for performing association rule mining to discover frequent patterns. Features like predefined support count, type of dataset, the size of the dataset (Either it has less number of items or high) etc.

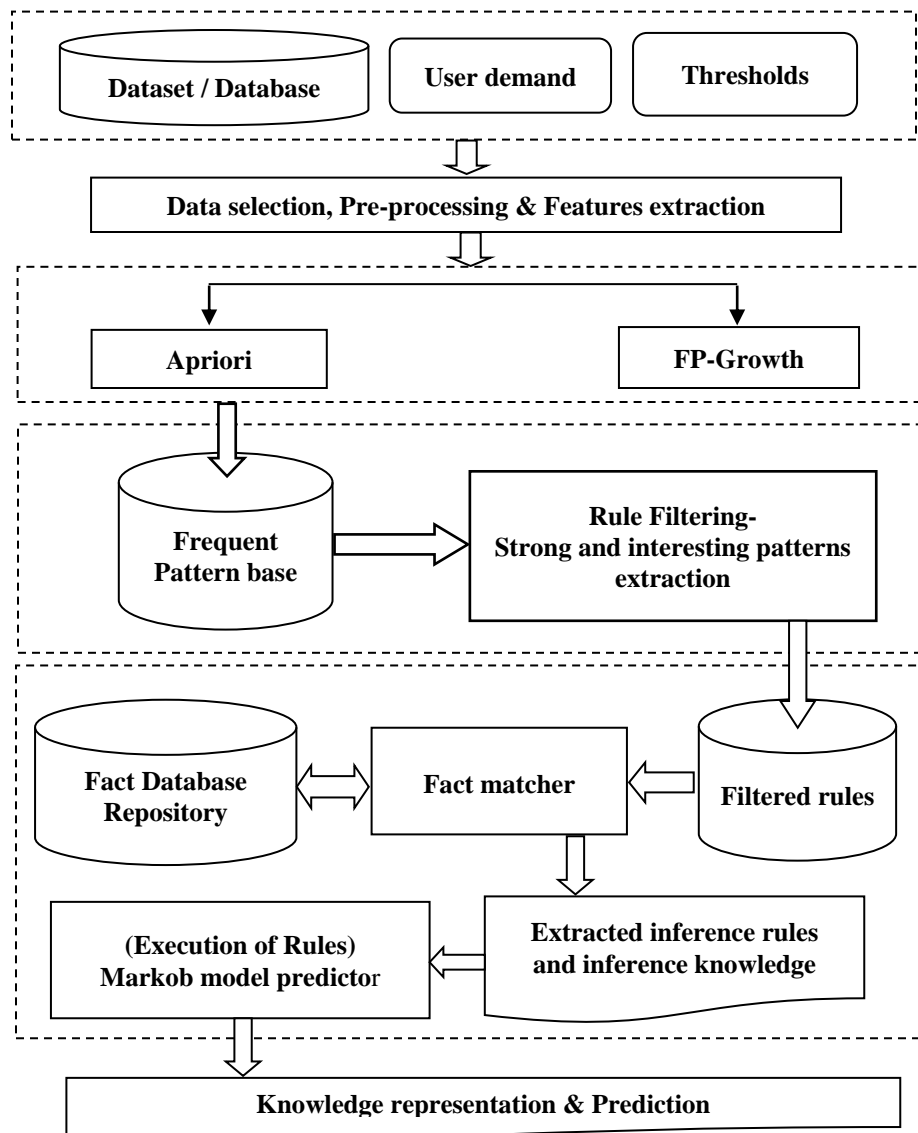


Fig. 4. Inference Mechanism Framework

B. Selection of ARM Approach based on features of data:

According to the analysis of first phase it selects the most appropriate approach for efficiently performing association rule mining, these selection criteria are as follows: Apriori Algorithm - Outperform when predefined support count is high and number of items are less (If size of dataset is low). FP Growth – outperform when the low support count is given and fast results are needed. It requires much storage space in case large transaction set is given to design and store a tree structure. Discovered rules are stored in rule base which further filters in next step by applying filtering and strength checking techniques. The procedure is as follows:

**If (Is High (Support) && Is High (Confidence) && Is Large (DB_ Size)) Then
 RB = Apriori (Dataset)
 Else**

RB = FP_Growth (Dataset)

C. Rule filtering & strength checking:

Phase 2 generates a huge number of generalized association rules stored in the rule base. In this phase the rule filtering and strength checking techniques have been adopted to find interesting rules, the process of rule filtering is as follows:

Let K_Rule is a knowledge base and Rule [i] is an array of rules where N is the number of rules then.

**For (j=1 to Count)
 RI = Calculate Relate Intensity (K_Base (FRule [j]))
 If (RI [j] >= Required Intensity) Then
 FR[j] = FRule[i] //Where FR is final rules
 Else
 Discard (FRule[j])**

End if

Next

D. Fact matching & generating inference rules:

In the fourth phase, we adopt forward chaining technique to discover inference rules. In this inference function process the filtered rule base by matching rules to the given facts (fact repository) to derive new knowledge (inference rules), it uses reasoning by matching and unification of similarity between the objects. In the process the pattern matcher matches the filtered rules with the available facts in fact database repository, if the rule matches with the fact then rule will be selected as inference rule and consequent part of the fact will be fetched from the respective fact and will store in inference rule database. The procedure is as follows:

For (Each Fact Domain Knowledge)

If (Matches(FR[i], Domain Knowledge)) Then

Infr_Rule=FR[i] + "-" + Fact[i]

End If

Next

E. Results & prediction:

This phase adopted the Markov Model predictor to predict what will happen in the future using the inference rules. The Markov theory was first introduced by a Russian mathematician Andrey Markov [3] and gave the concept of Discrete Time Markov Chain (DTMC) [3, 20], Markov chain term refer to the sequence of linked random objects (represented in the form of states) with respective probability of occurred events over each other, where the prediction of next happening depends on the current state of the system. Formally, it often represented as the form of directed graph where each event is represented as a state and weight of edges are represented as occurred probability of states as shown in Figure 5.

Let S be the set of distinct states $S = \{S_1, S_2, S_3, \dots, S_n\}$ and P is the set of distinct probabilities when the state takes moves from one state to other $P = \{P_1, P_2, P_3, \dots, P_n\}$, Here state (S) denoted as S_T at different time slot T.

$$P(S|A, \pi) = P(S_1) P(S_2|S_1) P(S_3|S_2 S_1) \dots P(S_T|S_1 \dots S_{T-1})$$

$$P(S | A, \pi) = P(S_1) P(S_2|S_1) P(S_3|S_2) \dots P(S_T|S_{T-1})$$

Each state of hidden Markov model is associated with probabilistic function so if O_t is the observation at time 't' generated by probabilistic function F Then $F_i = P(O_t|S_{t=i})$.

If N states $S_1, S_2, S_3, \dots, S_N$ are involved in the process, then the Markov chain would be represented as follows.

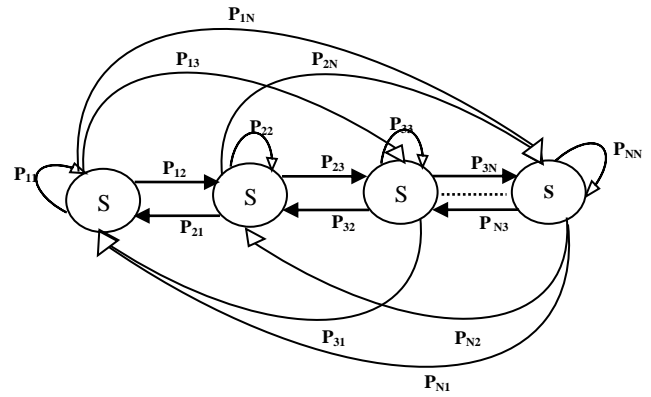


Fig. 5. Markov Chain for 'N' distinct states

Above transitions diagram can be represented in the form of following transition probability matrix.

$$\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ \dots \\ S_N \end{bmatrix}_{n+1} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & \dots & \dots & P_{1N} \\ P_{21} & P_{22} & P_{23} & \dots & \dots & P_{2N} \\ P_{31} & P_{32} & P_{33} & \dots & \dots & P_{3N} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ P_{N1} & P_{N2} & P_{N3} & \dots & \dots & P_{NN} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ \dots \\ S_N \end{bmatrix}_n$$

If we consider $IP_1, IP_2, IP_3, \dots, IP_N$ are the initial probabilities and the initial state transition matrix is S_0 then $S_i = [IP_1 \ IP_2 \ IP_3 \ \dots \ IP_N]$ Where transition probability matrix is

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & \dots & \dots & P_{1N} \\ P_{21} & P_{22} & P_{23} & \dots & \dots & P_{2N} \\ P_{31} & P_{32} & P_{33} & \dots & \dots & P_{3N} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ P_{N1} & P_{N2} & P_{N3} & \dots & \dots & P_{NN} \end{bmatrix}$$

So by multiplying identity matrix with probability matrix for Markov prediction, equation will be as follows $S_{i+1} = S_i \times P$ Here maximum outstanding probability in between $S_i = [IP_1 \ IP_2 \ IP_3 \ \dots \ IP_N]$ will be responsible for next most probable event.

Algorithm: Association Inference Rule Miner (abbreviated as AIRM)

Abbreviations Used:

- FR Final Rule
- FRule Filtered Rule
- EF Extracted Features
- RI Related Intensity
- RB Rule Base
- Infr_Rule Inference Rule Predictor
- IR Inference Rule Prediction
- TM Transition matrix

INPUT : Min_Support(S), Min_Confidence(C) (Thresholds), User_Demand(If Any), Dataset Data Base.

OUTPUT: Inference Rules, Results & Future Prediction

Begin:

- 1) EF = Extract Features(DataSet)
- 2) If (Is High(EF(Support)) && Is Large(DB_Size)) Then
 - a) RB = Apriori(Dataset)
- // Agrawal R. and Srikant R [1]
- Else
 - b) RB = FP_Growth(Dataset)
- // Han J., Pei J., and, Yin Y [10]
- 3) End If
- 4) For (j=1 to Count)
- 5) RI = Calculate Relate Intensity (K_Base(FRule[j]))
- 6) If (RI[j] >= Required Intensity) Then
 - a) FR[j] = FRule[i]
- 7) Else
 - b) Discard (FRule[j])
- 8) End if
- 9) Next
- 10) For (Each Fact Domain Knowledge)
 - a) If (Matches(FR[i], Domain Knowledge)) Then
 - i. Infr_Rule[i]=FR[i] + "-" + Fact[i]
 - b) End If
- 11) Next
- 12) Markov Inference Predictor (Infr_Rule[i])
- 13) End

End

Function: Markov Inference Predictor (Infr_Rule[i])

- 1) N = Count number of inference rules
- 2) For j=1 to N
 - a) P[i]=Calculate Probability of Facts[i]
- //Add P[i] in transition probability matrix
- b) TM= Matrix (P[i])
- 3) Next
- 4) Count=Cont_Column(TM)

- 5) S[0]=[Identity Matrix of Size (1× Count)]
- 6) While (S[i]! = S[i+1])
 - a) S[i+1]=s[i]*TM
 - b) i++

End do

VI. AN EXAMPLE

For example, we use a medical dataset (patient's symptoms information) of a city to predict which disease commonly affects a city. For this purpose, we propose an inference mechanism framework for association rule mining in this, firstly it identifies most suited ARM algorithm on the basis of features (thresholds, object's type and size of the dataset) of giving data set in Table-1 to find association rules.

TABLE I. PATIENT'S DATASET

Patients	Symptoms			
	Fever	Chills	Headache	-
1	Fever	Chills	Headache	-
2	Fever	Joint pain	Headache	-
3	Fever	Chills	Headache	-
4	Fever	Chills	Headache	-
5	Fever	Joint pain	Headache	-
6	Fever	Chills	Sweats	-
7	Fever	Joint pain	Headache	-
8	Fever	Chills	Headache	-
9	Fever	Joint pain	Headache	-
10	Fever	Muscle	Headache	joint pains
11	Fever	Chills	Headache	-
12	Fever	Joint pain	Headache	-
13	Fever	Chills	Headache	-
14	Fever	Muscle	Headache	joint pains
15	Fever	Muscle	Headache	joint pains

Table-2 shows resulting non redundant high intensity association rules generated by ARM algorithm are stored in knowledge.

TABLE II. RESULTING ASSOCIATION RULES

Patients	Symptoms			
	Fever	Chills	Headache	-
1, 3, 4, 8, 11, 13	Fever	Chills	Headache	-
2, 5, 7, 9, 12	Fever	Joint pain	Headache	-
10, 14, 15	Fever	Chills	Headache	joint pains

Fact repository shown in Table-3, if the antecedent matches with rules of Table-2 then rule are fixed and its fact is executed and commit as inference process is shown in Table-3. In the early stages of Malaria, Viral fever, Chikungunya and Dengue fever symptoms are sometimes similar to these.

TABLE III. FACT DATABASE

ID	Antecedents	Fact
1	Fever, Chills, Headache	Malaria
2	Fever, Joint pain, Headache	Viral fever
3	Headache, Nausea, vomiting, Conjunctivitis, Maculopapular rash	Chikungunya

In this step inference function perform the backward chaining on the rules stored in the knowledge base and compare each resulting rule with fact stored in fact data base process is shown Table-4.

TABLE IV. MATCH RULES WITH FACTS

Pati ents	Symptoms				I D	Antecedents	Fact
1, 3,4, 8, 11, 13	Fe ver	Ch ills	Head ache	-	1	Fever, Chills, Headache	Malaria
2, 5, 7,9 12	Fe ver	Joi nt pain	Head ache	-	2	Fever, Joint pain, Headache	Viral fever
10, 14, 15	Fe ver	Ch ills	Head ache	Jo int pain s	3	headache, nausea, vomiting, conjunctivitis	Chikungunya

TABLE V. RESULTING INFERENCE RULES

Patients	Symptoms	Fact
1, 3,4, 8, 11, 13	Fever, Chills, Headache	Malaria
2, 5, 7,9 12	Fever, Joint pain, Headache	Viral fever

As per above calculation the probability of Malaria and Viral fever over each other is as follows:

TABLE VI. PROBABILITY CALCULATION

Fact	Support Count	Probability	
		Malaria	Viral fever
Malaria (M)	40%	0.55	0.45
Viral fever (V)	33%	0.45	0.55

Figure -6 shown the Markov chain, according to above given probabilities. Where Malaria (M) and Viral fever (V) are the states.

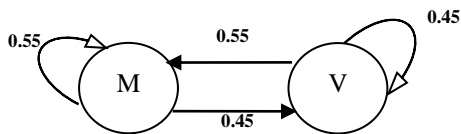


Fig. 6. Transition diagram (Markov Chain)

Figure-6's Markov chain can be represented in the form of the transition matrix (TM) as follows

$$TM = \begin{matrix} M \\ V \end{matrix} \begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{bmatrix}$$

Where the Identity Matrix is

$$(S_0) = [1 \ 0]$$

$$\text{From } S_{n+1} = S_n \times TM$$

$$S_1 = [1 \ 0] \begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{bmatrix} = [0.55 \ 0.45]$$

Now by multiplying S1 with transition probability matrix.

$$S_2 = [0.55 \ 0.45] \begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{bmatrix} = [0.5050 \ 0.4950]$$

So in next month the probability of Malaria or Viral fever is 0.5050 and 0.4950 respectively.

$$S_3 = [0.5050 \ 0.4950] \begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{bmatrix} = [0.5005 \ 0.4995]$$

After two month the probability of Malaria or Viral fever is 0.5005 and 0.4995 respectively.

$$S_4 = [0.5005 \ 0.4995] \begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{bmatrix} = [0.5001 \ 0.5000]$$

After three month the probability of Malaria or Viral fever is 0.5001 and 0.5000 respectively.

$$S_5 = [0.5001 \ 0.5000] \begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{bmatrix} = [0.50 \ 0.50]$$

After four month the probability of Malaria or Viral fever is 0.50 and 0.50 respectively.

$$S_6 = [0.5000 \ 0.5000] \begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{bmatrix} = [0.50 \ 0.50]$$

After five month the probability of Malaria or Viral fever is 0.50 and 0.50 respectively.

Therefore, it would appear that after 4 months, approximately 50% of patients in the city are more likely to get prone to viral disease. On the other hand, approximately 50% of patients may get prone to Malaria.

VII. RESULT AND DISCUSSION

In this inference approach we are examining a sample database of a city hospital, it contains patients' symptoms information as shown in figure-7, where a MATLAB based Inference System is used to match facts with inference rule to identify disease.

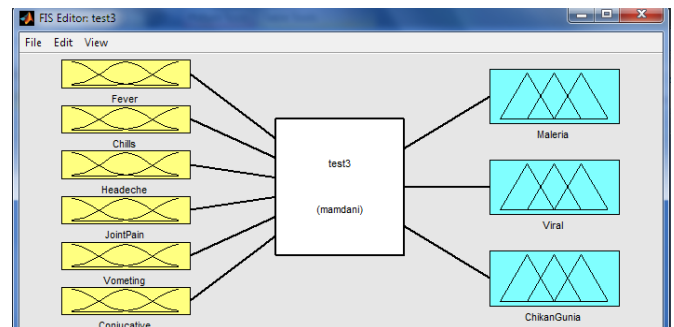


Fig. 7. Inference System in MATLAB

Algorithm AIRM accept this data as input and analyzes, after association inference rule mining it provides some inference knowledge as a result, which is represented in the figure-7, histogram shows that, according to the observations, after 4 months the disease probability does not change over time, it becomes steady. Eventually, continuing to multiply our answer by the transition matrix again and again, has no effect. So we can infer that the data can be used full for prediction up to 4 months. In Figure-8 Y-axis and X-axis have shown the probability and months respectively.

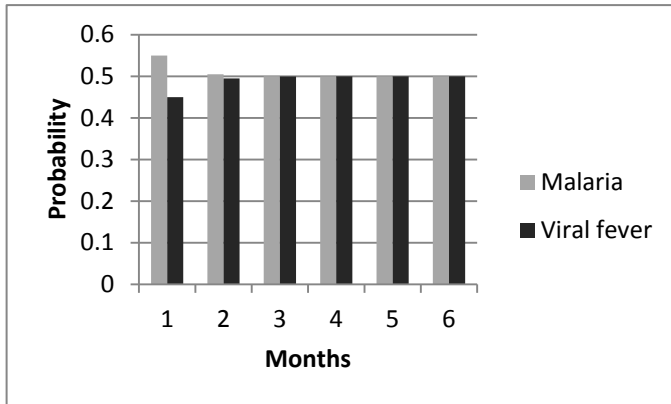


Fig. 8. Probability comparative study

VIII. CONCLUSION

In this paper, we have proposed Markov model based inference framework for association rule mining, in first tier we adopted most suitable ARM techniques to find the frequent and interesting patterns. In second tier it checks the strongest rules as well as removes redundant and trivial rules in order to increase efficiency, and find the inference rules by applying forward chaining inference technique.

In tier-3 Markov predictor accepts these inference rules with their respective probabilities to predict about future possibilities. This approach would work as pave for future research because that approach can be used in weather forecasting, medical disease prediction and stock market prediction etc.

REFERENCES

[1] Agrawal R. and Srikant R., "Fast algorithms for mining association rules", Proceedings 20th Very Large Databases Conference, Santiago, Chile, pp.487-499, 1994.
[2] Agrawal R., Imielinski T., and Swami A., "Mining Association Rules between Sets of Items in Large Databases", In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 207-216, 1993.
[3] Andrew M., "Hidden Markov Models", Andrew's tutorials. www.autonlab.org/ tutorials/.

[4] Chi X., "A New Matrix-Based Association Rules Mining Algorithm", 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012), IEEE, pp. 633-636, 2012
[5] Cooper B., Lipsitch M., "The analysis of hospital infection data using hidden Markov models", Biostatistic, Printed in Great Britain, pp. 223-237, 2004.
[6] Fagin R., Joseph, Halpern Y. and Vardi Moshe Y., "What is an Inference Rule?" The Journal of symbolic logic, Vol. 57. Number 3. Sept., pp. 1018-1045, 1992.
[7] Fayyad Usama M., Piatetsky-Shapiro G., Smyth P. "From data mining to knowledge discovery: an overview", Advances in knowledge discovery and data mining, American Association for Artificial Intelligence, ACM, pp.1 - 34, 1996.
[8] Gautam P., Pardasani K. R., "A Fast Algorithm for Mining Multilevel Association Rule Based on Boolean Matrix", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 03, pp 746-752, 2010.
[9] Han J., Kamber M., "Data Mining Concepts and Techniques", Morgan Kaufmann Publishers, San Francisco, USA, ISBN 1558604898, 2001.
[10] Han J., Pei J. and, Yin Y.. "Mining frequent patterns without candidate generation", In Proceedings of the 2000 ACM SIGMOD international conference on Man agement of data, ACM, pp 1-12, 2000.
[11] Han J., Fu Y., "Discovery of multiple-level association rules from large databases. Proceedings of the VLDB Conference", pp-420-431, 1995.
[12] <http://jena.apache.org/>, retrieved on 20th September 2013.
[13] <http://sweetrules.semwebcentral.org/>, retrieved on 22nd September 2013.
[14] <http://www.jessrules.com/jess/download.shtml>. retrieved on 5th January 2013.
[15] <http://www.vistology.com/basevisor/basevisor.htm>, retrieved on 9th February 2013.
[16] Khare N., Adlakha N., Pardasani K. R., "An Algorithm for Mining Multidimensional Association Rules using Boolean Matrix", IEEE International Conference on Recent Trends in Information, Telecommunication and Computing, pp. 95-99, 2010.
[17] Lee C., Chen M., Lin C., "Progressive Partition Miner: An Efficient Algorithm for Mining General Temporal Association Rules", IEEE Transactions on knowledge and data engineering, Vol. 15, No. 4, pp.1004-1014, 2003.
[18] Liu H. and Wang B., "An Association Rule Mining Algorithm Based On Boolean Matrix", Data Science Journal, Volume 6, pp-63-66, 2007.
[19] Pav'on J., Viana S., and G'omez S., "Matrix apriori: Speeding up the search for frequent patterns". In Proceedings of the 24th IASTED International Conference on Database and Applications, DBA'06, Anaheim, CA, USA ACTA Press, pp. 75-82, 2006.
[20] Rabiner L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proc. of the IEEE, Vol.77, No.2, pp.257-286, 1989.
[21] Venus S. , Akram M. , and Abeer A. , "Arabic Expert System Shell", IAJIT First Online Publication, Vol.10, no.1, 4094-6, 2011.
[22] Yang J., Liu J., Wang J., Sii H., and Wang H.,Wei, "Belief Rule-Based Inference Methodology Using the Evidential Reasoning Approach—RIMER", IEEE Transactions on systems, man and cybernetics part A: System and Humans, Vol. 36, No. 2, pp. 266-285, 2006.
[23] Yuan Y., Huang T., "A Matrix Algorithm for Mining Association Rules", Springer-Verlag Berlin Heidelberg, pp-370-379, 2005.
[24] Zhang Z., Zhang Y., "A parallel algorithm of frequent itemsets mining based on bit matrix", International Conference on Industrial Control and Electronics Engineering, pp.1210-1213, 2012.