ISSN : 2165-4069(Online)
ISSN : 2165-4050(Print)

IJARAI

International Journal of Advanced Research in Artificial Intelligence

Volume 3 Issue 9

www.ijarai.thesai.org

A Publication of
The Science and Information Organization

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ARTIFICIAL INTELLIGENCE

THE SCIENCE AND INFORMATION ORGANIZATION

www.thesai.org | info@thesai.org

Editorial Preface

From the Desk of Managing Editor…

"The question of whether computers can think is like the question of whether submarines can swim." — Edsger W. Dijkstra, the quote explains the power of Artificial Intelligence in computers with the changing landscape. The renaissance stimulated by the field of Artificial Intelligence is generating multiple formats and channels of creativity and innovation.

This journal is a special track on Artificial Intelligence by The Science and Information Organization and aims to be a leading forum for engineers, researchers and practitioners throughout the world.

The journal reports results achieved; proposals for new ways of looking at AI problems and include demonstrations of effectiveness. Papers describing existing technologies or algorithms integrating multiple systems are welcomed. IJARAI also invites papers on real life applications, which should describe the current scenarios, proposed solution, emphasize its novelty, and present an in-depth evaluation of the AI techniques being exploited. IJARAI focusses on quality and relevance in its publications.

In addition, IJARAI recognizes the importance of international influences on Artificial Intelligence and seeks international input in all aspects of the journal, including content, authorship of papers, readership, paper reviewers, and Editorial Board membership.

The success of authors and the journal is interdependent. While the Journal is in its initial phase, it is not only the Editor whose work is crucial to producing the journal. The editorial board members , the peer reviewers, scholars around the world who assess submissions, students, and institutions who generously give their expertise in factors small and large— their constant encouragement has helped a lot in the progress of the journal and shall help in future to earn credibility amongst all the reader members.

I add a personal thanks to the whole team that has catalysed so much, and I wish everyone who has been connected with the Journal the very best for the future.

Thank you for Sharing Wisdom!

Editor-in-Chief
IJARAI
Volume 3 Issue 9 September 2014
ISSN: 2165-4069(Online)
ISSN: 2165-4050(Print)

Editorial Board

Reviewer Board Members

CONTENTS

An Inference Mechanism Framework for Association Rule Mining

Kapil Chaturvedi
Department of Computer Application
Rajiv Gandhi Proudyogiki
Vishwavidyalaya
Bhopal, MP, India

Dr. Ravindra Patel
Department of Computer Application
Rajiv Gandhi Proudyogiki
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, $I_{3,}$……., I_n which may occur in different transactions, I= {I_1, I_2, $I_{3,...}$……., I_n}.

Definition 2: Let T is set of transactions contains different transactions t_1, t_2, t_3,…., t_m: T= {t_1, t_2, t_3,…., 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→Y is said to be an association rule, if the minimum support count S ($X \rightarrow Y$) $\geq$ Min (S) and minimum confidence C (X→Y)$\geq$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∪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∪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)"

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

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.

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

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) $P(X \not\wedge Y) = P(X) \times P(Y)$ – *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

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

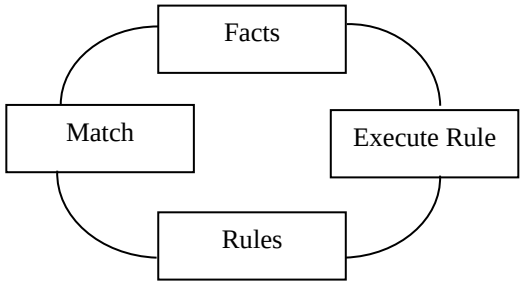


Fig. 1. Traditional inference approach

- ***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

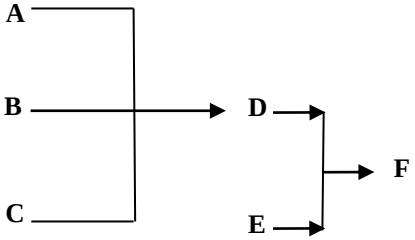


Fig. 2. Search strategy of forward chaining system

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

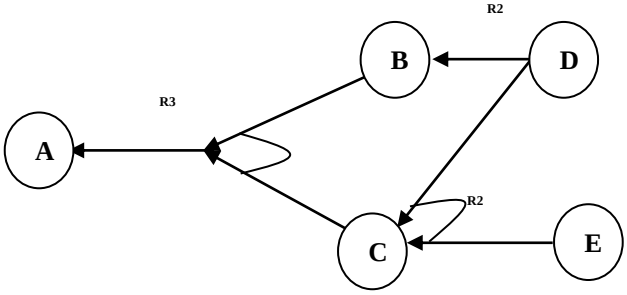
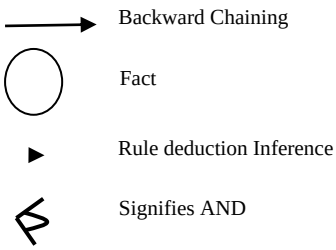


Fig. 3. Search strategy of backward chaining system



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 at 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 generates 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 at al. proposed an approach generic rule-base 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 Channing 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 & Featuress 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.

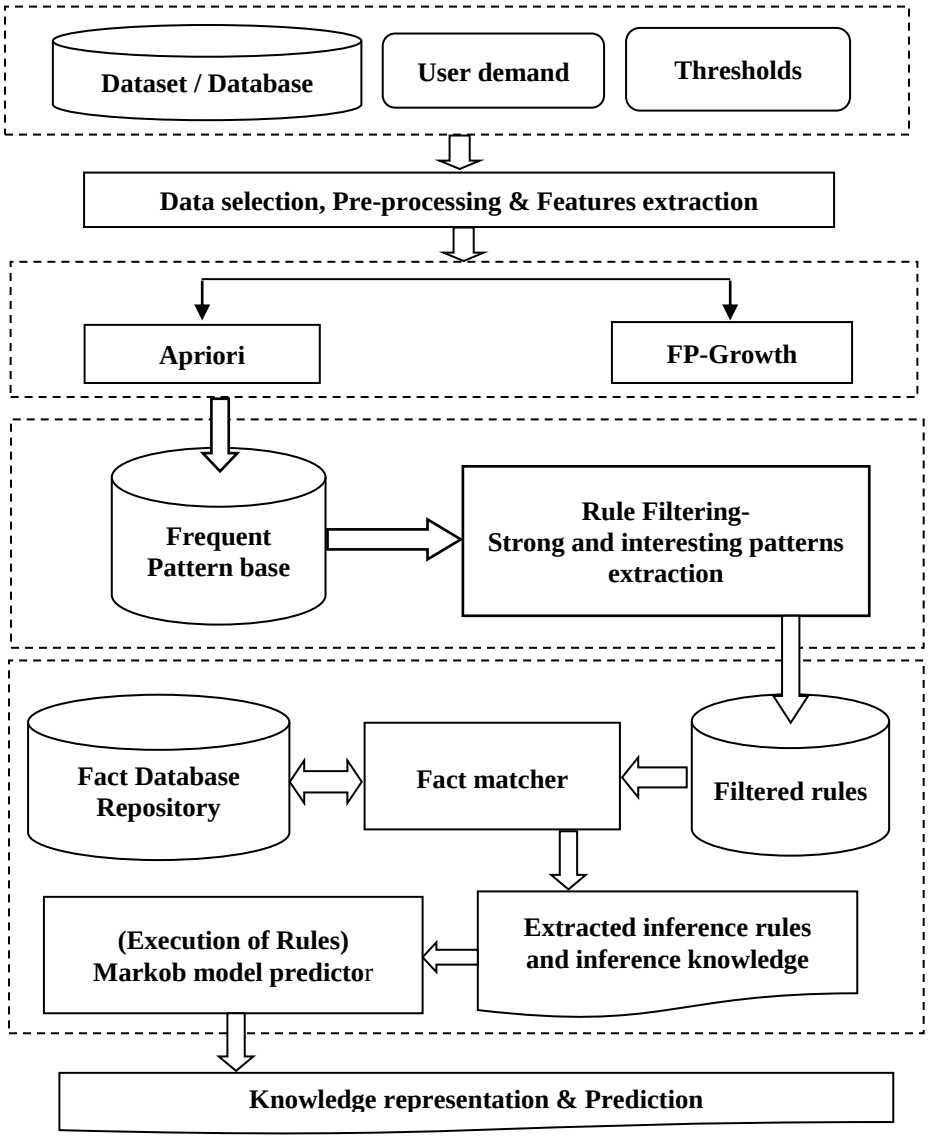


Fig. 4. Inference Mechanism Framework

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…… 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…… P_n}, Here state (S) denoted as S_T at different time slot T.

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

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…… S_N are involved in the process, then the Markov chain would be represented as follows.

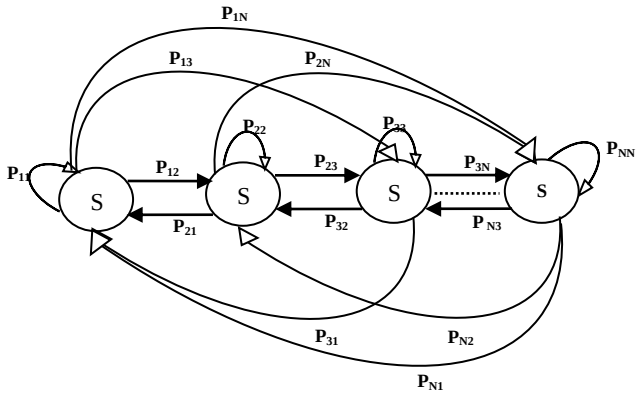


Fig. 5. Markov Chain for 'N' distinct states

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

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

If we consider IP_1, IP_2, $IP_{3\ldots\ldots}$ IP_N are the initial probabilities and the initial state transition matrix is S_0 then $S_i = \begin{bmatrix} IP_1 & IP_2 & IP_3 & \ldots & \ldots & IP_N \end{bmatrix}$ Where transition probability matrix is

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

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 $S_i = \begin{bmatrix} IP_1 & IP_2 & IP_3 & \ldots & \ldots & IP_N \end{bmatrix}$ 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			
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			
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

Patients	Symptoms			
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

⇔

ID	Antecedents	Fact
1	Fever, Chills, Headache	Malaria
2	Fever, Joint pain, Headache	Viral fever
3	headache, nausea, vomiting, conjunctivitis	Chikunguny a

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.

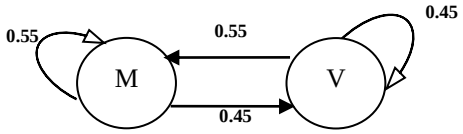


Fig. 6. Transition diagram (Markov Chain)

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

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

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

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

Now by multiplying S1 with transition probability matrix.

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

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

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

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

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

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

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

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

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

Aftere 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.

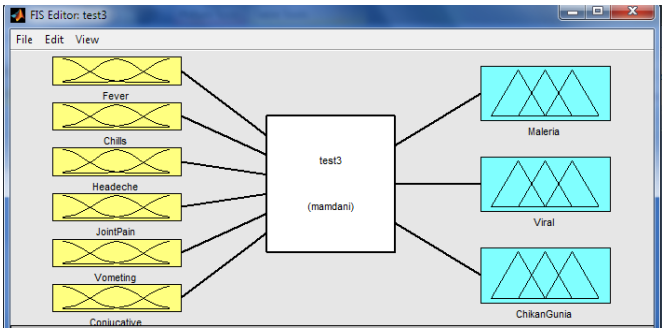


Fig. 7. Inference System in MATLAB

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.

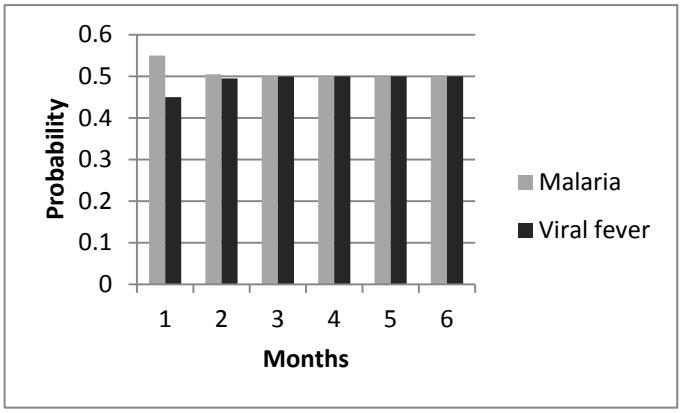


Fig. 8. Probability comparative study

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 Jourxal 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-Base 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.

Fuzzy Concurrent Object Oriented Expert System for Fault Diagnosis in 8085 Microprocessor Based System Board

Mr.D.V.Kodavade[1]
Phd. Student, Electronics Engg,
Shivaji University
Kolhapur

Dr. Mrs.S.D.Apte[2]
Professor in Electronics Engg.
Rajashree Shahu College of Engineering,
Pune

***Abstract*—With the acceptance of artificial intelligence paradigm, a number of successful artificial intelligence systems were created. Fault diagnosis in microprocessor based boards needs lot of empirical knowledge and expertise and is a true artificial intelligence problem. Research on fault diagnosis in microprocessor based system boards using new fuzzy-object oriented approach is presented in this paper. There are many uncertain situations observed during fault diagnosis. These uncertain situations were handled using fuzzy mathematics properties. Fuzzy inference mechanism is demonstrated using one case study. Some typical faults in 8085 microprocessor board and diagnostic procedures used is presented in this paper.**

Keywords—Expert Systems; fuzzy; Inference; Knowledge base

I. INTRODUCTION

The goal of Artificial Intelligence (AI) is to design and generate computer programs which exhibit some features of human intelligence. Artificial intelligence is defined as the ability to acquire, understand and apply knowledge. AI has number of important sub areas like expert systems, natural language processing, computer vision, theorem proving, game playing, robotics etc. The most successful area of artificial intelligence is expert systems. Expert systems are used for complex problem solving and are having number of successful applications in industries. One of the important unattempted application is fault diagnosis in electronic circuits. Fault diagnosis methodology operates on observed erroneous behavior and hardware structure of the unit under test. The erroneous behavior consists of responses of different components on the output lines on specific input values. Present research work relates to artificial intelligence systems and more particularly to fault diagnostic expert system using fuzzy object oriented approach. The basic components of expert systems are knowledge base, inference engine and user interface. The paper discusses the implementation of knowledge base, inference mechanism and user interface and also explores an innovative strategy developed for fault diagnosis.

II. LITERATURE REVIEW

Yan Qu et al. [11] discussed fuzzy diagnostic expert system for electric control engine. Commix fuzzy reasoning method is used in inference engine. Proposed expert system includes knowledge base, reasoning machine, explain system, management system and human machine interface modules. An intelligent fault diagnosis framework based on fuzzy integrals is built by M. Karakose et al. [12]. The method consists of two frameworks. The first framework is used to identify the relation between features and a specified fault and the second framework integrates different diagnostic algorithms to improve the accuracy rate. Approach is experimented on 0.37 KW induction motor, where broken rotor bar and stator faults were evaluated to validate the model.

Liang Xiao-lin, et al. [13] introduced fuzzy set theory into electronic fault tree analysis and scientifically analyzed the various kinds of fuzzy information confronted by the failure of the electronic equipment. The authors stated that the method can analyze and process random uncertainty and fuzzy uncertainty failure simultaneously and can efficiently solve the problem of electronic equipment fault diagnosis.

Zhang Chao Jie, et al. [14] has suggested an ant colony algorithm for test point selection of analog circuits based on fuzzy theory. Authors discussed use of this algorithm for fault diagnosis in time delay circuit boards used in marine engine.

Zhiyong Wang, et al. [15] presents a rough set based fuzzy logic technique which diagnoses multiple faults in a transformer by applying rough fuzzy set theory to the International Electro technical Commission (IEC) codes. By using fuzzy method, the fuzzy membership functions of every fault diagnosis decision rules are displayed and finally the fault type of the transformer is diagnosed.

Jiang-Liang Chen, et al. [16] built a fuzzy expert system for fault diagnosis in electric distribution system. Based on the symptoms description derived from customers and historical trouble tickets information the system determines the membership grade. The membership grade indicates the degree to which specific component might be faulty on prioritized basis. To demonstrate the effectiveness of the approach, the system is applied to practical data which includes 3067 trouble tickets.

Qu Yan[11] developed a fuzzy expert system framework using object oriented technique. Knowledge base is developed by organizing rules and facts in to different object groups respectively. Facts objects uses object oriented concepts like inheritance, encapsulation & polymorphism. The rule objects contain several specific components to process fuzzy

information .Fuzzy set approach is only used in rule base for organizing the rule base. The traditional backward chaining inference engine is implemented by authors.

From the literature survey it is observed that, many strategies developed uses rule based approach for fault diagnosis. Fuzzy object oriented approach is unattempted for fault diagnosis in processor based system boards. There are many drawbacks of rule based approach as per literature survey. The problems associated with rule based approaches may be solved using object oriented paradigm. In object oriented design classes form hierarchical structures and hence may encapsulate the items easily. Since objects can communicate with each other by message passing, the search in large databases may become easy. At the run time the member functions associated with the classes can store the arguments. This may reduce the need for a working memory. Concurrency in testing digital components can be achieved using multithreading. All these issues are discussed following subsections.

III. Architecture of Fuzzy Object Oriented System

The architecture of the new approach is shown in Figure1. It consists of fuzzy object oriented knowledge base, fuzzy inference mechanism using message passing, user interface and working memory to store facts.

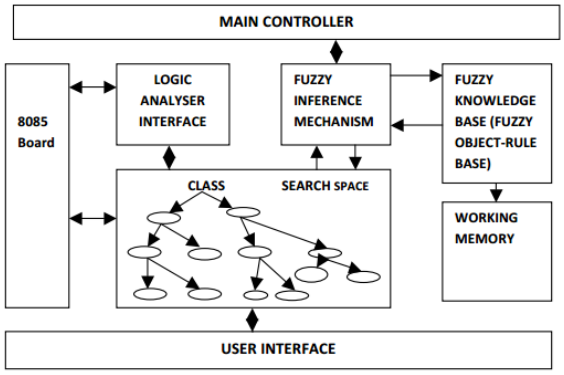


Fig. 1. Architecture of Fuzzy Object Oriented System

As shown in Figure 1, the main controller module controls entire system. Fault diagnostic knowledge is stored in knowledge base. Fuzzy properties are incorporated in knowledge representation, inference mechanism in order to complete fault diagnosis process more naturally. The fuzzy object diagnostic model is formed by interconnection of objects in the object space. The object space is the part of the working memory where objects are fired at run time. Every object is associated with fuzzy membership values. The knowledge base consists of procedural knowledge and declarative knowledge. Procedural knowledge is implemented using fault diagnostic procedures for different faults in 8085 microprocessor board. Procedural knowledge is acquired from domain experts and by actual working on board and is represented using fuzzy-object-rules. Declarative knowledge consists of knowledge about component connectivity on 8085 microprocessor board and is represented using frame structures. Inference mechanism is implemented using new message passing algorithm with forward chaining. In knowledge base and inference mechanism inexactness is handled by using fuzzy membership values. These values are assigned by domain experts and used as heuristic functions for guiding search process. For interaction with maintenance technician graphical user interface is developed. The responses of components for different test cases are stored temporary in working memory. The detailed implementation of knowledge base and inference mechanism is explained in next subsection.

A. Fuzzy Knowledge Base

The procedural knowledge is categorized as fault isolation knowledge and check knowledge. Fault isolation knowledge is used to isolate the fault area. Classes and methods are used to represent this type of knowledge. Fault isolation knowledge is implemented using Fault isolate method which is invoked by CFault_diagnose_fuzzyq classs constructor at run time. The fuzzyq represents fuzzy quantification value for specified constructor. The fault isolate method returns suspected faulty component with fuzzy quantification values. The procedural knowledge is implemented using diagnose method under different component classes. The Ccomponent class is used for writing diagnostic procedures for specific component on board. Each component class is associated with fuzzy membership value. compare( ) method is implemented to compares the fuzzy confidence value of the faulty components with threshold value. The membership value 0.9 means most confident while membership value 0.1 means less confident. “Is_Ok” and “Number_methods” flags are implemented to count the diagnosed faulty components on board and number of methods invoked.

The typical class diagram for procedural knowledge representation is shown in Figure 2. CFault_diagnose_fuzzyq class contains Ccomp_ fuzzyq object as a data member (composition /relation). Ccomp_fuzzy is a base class for all specific components like Ccomp_8085_fuzzyq, Ccomp_8255_fuzzyq etc. The concrete classes overrides the diagnose method from different component classes.

Similarly, procedural knowledge is represented for all the components available on the board. The declarative knowledge is used to describe the interconnections of components and is implemented using frame structures as discussed in the previous chapter. This knowledge is used in inference process as well as to train the maintenance technician by providing guidance on component connectivity on board. Concurrency is handled using multithreading class.

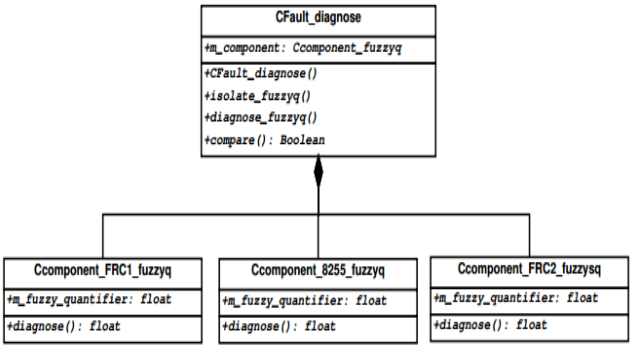


Fig. 2. Implementation of Knowledge Base

B. Fuzzy Inference Mechanism

Inference mechanism module works under main controller. The basic goal of this module is, to search optimistically in knowledge base for deciding diagnostic strategy for specified fault. For search in knowledge base it uses message passing mechanism modified with fuzzy quantifiers.To handle uncertainty in fault diagnosis and to implement expert judgments fuzzy quantification values are used. The fuzzy inference mechanism computes the degree of confidence in the conclusion that specified component is faulty. This confidence value lies in the range 0.1 to 0.9. The system adjust the most promising faulty component by providing confidence value close to 0.9. Every diagnosed faulty component is associated with confidence value. The fuzzy inference forward chaining algorithm computes the confidence value by taking minimum of all fuzzy quantifiers associated with methods connected by AND operator (intersection property) under one component class and multiplying it with fuzzy quantifier value of respective component class.

As the inference mechanism works using message passing it tries to get all possible faulty components for the specified fault from the object tree. To limit the list of suspected faulty components a threshold is used. Threshold value is decided by the maintenance technician as per the complexity of the fault and obtains most promising faulty components.

IV. Strategy for Fault Diagnosis

Figure 3 illustrates the fault diagnostic strategy using sequence diagram. Fault diagnosis is carried out in two phases. In the first phase the maintenance technician selects the fault query from the list. After the fault is identified an object instance is created and controller invokes CFault_diagnose_fuzzyq constructor class. It initializes all flags and invokes the primary check methods. The methods pass arguments to the maintenance technician and get values in Boolean form and return the suspected faulty component to the constructor with fuzzy quantifications and thus isolate the fault area. Constructor updates the member variable associated with it.

In the second phase, diagnose method invokes identified component subclass having fuzzyq value close to 0.9. The specified component concrete class calls diagnostic methods and carry out tests by passing arguments to the maintenance technician. He responds by providing status of IC pins. In many cases in the present system, an apparent problem in one part of the circuit is actually caused by a fault in the related part of the circuit. As a result, when trying to diagnose the cause of set of symptoms for one component, the system search for related symptoms and faults with other components by message passing. Thus systems give all possible faulty components having common symptoms with fuzzy confidence values. Here to get most promising faulty component value fuzzy confidence value is used. The component having more confidence value i.e. closer to 0.9 is considered as most promising faulty component. The output of the second phase is a list of most likely faulty components derived by the system. As per the threshold value selected by technician the faulty components are displayed in diagnosed fault list with fuzzy confidence values. The technician can get remaining less probable faulty components by selecting lower threshold value.

Concurrency in fault diagnosis is implemented using multithreading approach.

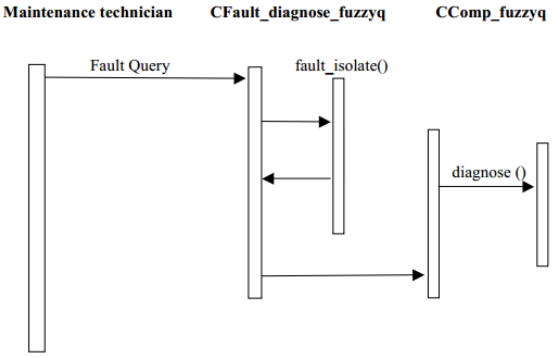


Fig. 3. Fault Diagnosis Strategy

V. Fault Diagnosis in 8085 Microprocessor Based System Board

The 8085 microprocessor board is taken as a unit under test for fault diagnosis. About 65 different faults were identified and diagnosed using this new fuzzy object oriented fault diagnostic system. One fault and the diagnostic strategy applied by new approach is discussed in next subsections.

Experiment 1 : Fault Query: No data get written form C100 H onwards.

On selection of this fault query from the menu, system generates an object instance. After pressing start diagnosis button from the menu the controller calls the fault diagnose class. For initialization and primary checks constructor is used. The constructor calls diagnose method form the generated object instance. Since there is possibility that, IC 8085 is faulty or may be IC_6116_U4 faulty. The diagnose method passes arguments *Does pin 20 U1 high?* and *Does clock present between pin 1_2of_U4?* to the controller both are uncertain for this particular fault. Which is to be tested first is decided by controller based on fuzzyq value. In the present approach the controller based on fuzzy quantification value passes diagnose method to check clock between pin 1 & 2 of 8085 processor as a first check. The technician responds “yes” to first arguments and also “yes” to second argument. The diagnose methods returns *U6_74ls138* and *U1_8085* to fault diagnose class as suspected faulty components with fuzzy quantification value. For the present fault fuzzy quantification value is more for U1_8085. The fault diagnose class invokes CComp8085fuzzyq class.The CComp_8085_fuzzyq class calls diagnose method under this class. The diagnose method passes *Does pin 20_U1_6264 low?* and receives “y” response from user and stores in memory. The next argument *Does Pin_37_U4_is_low?* Is passed to user and also receives “y” response. After getting both responses and stored responses under different classses the component class Ccomp_8085_fuzzyq returns *8085_U4* component as faulty component to fault diagnoses class with confidence value. The strategy is illustrated using class diagram in Figure 4. Here P

indicates test pass and F indicates test failed. Similarly all 60 faults were diagnosed and the results are shown in Table 1.

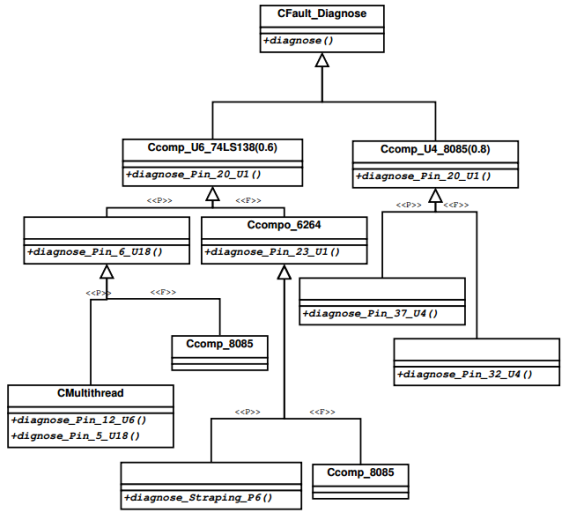


Fig. 4. Diagnosing Memory related fault

VI. Results Obtained During Fault Diagnosis in 8085Microprocessor Based System

After interacting with laboratory technicians and different industrial persons 60 commonly occurring faults for 8085 Microprocessor board are considered. Table1.shows the list of 10 typical faults and their diagnosis with confidence values obtained using fuzzy inference mechanism with number of methods required. Here threshold selected by the technician is 0.6, hence faults with confidence value more than 0.6 are displayed.

TABLE I. Results Obtained During Fault Diagnosis in 8085 Microprocessor Board.

Sr. No.	Fault Query	Using Concurrent Object Oriented Expert System.	Using Fuzzy Concurrent Object Oriented Expert System with Confidence Value (CV)
1	No sign on Message on reset pressed	EPROM 2732 Faulty No. of methods applied = 01	EPROM 2732 Faulty (0.8) No. of methods applied =01
2	Display D1 not working	74ls138_U7 faulty No. of methods applied = 01	Display D1 Faulty (0.7) No. of methods applied =01
3	Interrupt RST 6.5 not working	8085 faulty No. of methods applied = 02	Strapping P2 open (0.8) No. of methods applied =02
4	Memory read operation from C200H onwards not working	6264 faulty No. of methods applied = 02	6264 faulty (0.6) No. of methods applied =02
5	System not getting started	8085 socket failure No. of methods applied = 01	8085 socket failure (0.6) No. of methods applied =02
6	After reset key pressed system is not get resetting	Key in reset logic failure Resistor R6 in reset logic failure No. of methods applied = 02	8085 failure (0.7) No. of methods applied =02
7	On TURN ON display show garbage information	8279 faulty 2764 faulty No. of methods applied = 02	2764 failure (0.8) No. of methods applied =02
8	After execution of instruction MOVA,40H no register get modified with data	8279 faulty 8279 socket faulty No. of methods applied = 01	No fault diagnosed No. of methods applied =01
9	Program is not executed on pressing EXEC key	8279 faulty No. of methods applied = 01	EXEC key faulty (0.6) No. of methods applied =01
10	Data is not get written from C100 onwards	6264 Faulty 8085 faulty No. of methods applied = 01	6264 failure (0.7) No. of methods applied =01

VII. Conclusion

The fuzzy concurrent object oriented system diagnoses faults correctly. The use of fuzzy quantification in reasoning has provided inferencing in natural way. As compared with previous approach by incorporating fuzzy logic in knowledge base and in inference engine 60% faults were diagnosed using one method and 33.33% faults were diagnosed using two methods as compared with non fuzzy approach. From the results obtained it is concluded that, Fuzzy concurrent object oriented approach is superior than object oriented and rule based approach. The use of fuzzy set theory in reasoning has improved diagnostic efficiency by 30% using one method as against concurrent object oriented approach. By selecting threshold the most promising faults are only displayed to technicians and hence system diagnoses the faults more accurately and confidently than non fuzzy approach. The results are validated by industrial experts and are found correct.

References

[1] C. Angeli, “Diagnostic expert systems: From expert’s knowledge to real time systems,” International Journal of Advanced Knowledge Based Systems (Model, Applications & Search), vol. 1, pp. 50–70, 2010.

[2] D.N.Batanov and Z. Cheng, “An object-oriented expert system for fault diagnosis in ethylene distillation process,” Computer in Industry, Elsevier, vol. 27, pp. 237–249,Feb 2000.

[3] N. Yang and S. Zhang, “An expert system for vibration fault diagnosis of large steam turbine generator set,” in Proceeding of 3rd IEEE International Conference on Computer Research & Development ICCRD, vol. 2, pp. 217–221, 11-13 March 2011.

[4] J. W.Coffey and A. J. Canas, “Knowledge modeling and the creation of ei-tech: A performance support and training system for electronic technicians,” International Journal on Expert Systems with Applications, Elsevier, vol. 25, pp. 483–492, 2003.

[5] J. Qu and L. Liang, “A production rule based expert system for electronic control automatic transmission fault diagnosis,” in Proceeding of International Conference on Information Engineering and Computer Science, pp. 1–4, 2009.

[6] I. Borlea and A. Buta, “Diase-expert system fault diagnosis for timisoara 22kvsubstation,” inProceeding of Eurocon, pp. 251–255, 22-24 November 2005.

[7] C. Jingjie and C. Xiaxia, "Research on embedded airborne electronic equipment fault diagnosis expert system," in Proceeding of 2nd International Conference on Information Engineering and Computer Science, pp. 1–5, 2010.

[8] T. Han, B. Li, and L. Xu, "A universal fault diagnostic expert system based on bayesian network," inProceedings of IEEE International Conference on Computer Sc. & Software Engineering, pp. 260–265, 12-14 December 2008.

[9] S. Gebus and K. Leiviska, "Knowledge acquisition for decision support systems on an electronic assembly line,"International Journal on Expert Systems with Applications, Elsevier, pp. 94–101, 2007.

[10] C. Xu, Z. Xu, P. D. Xiao, Z. Zhou, and S. Liu, "An object-oriented expert system tool for fault diagnosis," in Proceeding of Instrumentation and Measurement Technology Conference,1995., pp. 614–617.

[11] Y. Qu, T. Fu, and H. Qiu, "A fuzzy expert system framework using object-oriented techniques," inProceeding of Pacific-Asia Workshop on Computational Intelligence and Industrial Application, vol. 2, pp. 474–477, 2008.

[12] M. Karakose, I. Aydin, and E. Akin, "The intelligent fault diagnosis frameworks based on fuzzy integral," in Proceeding of International Symposium on Power Electronics Electrical Drives Automation and Motion, pp. 1634–1639, 2010.

[13] L. Xiao-lin, Z. Yan-Xia, and Z. Zeng-hui, "Research on applications of fuzzy fault analysis in the electronics equipment fault diagnosis," in Proceedings of 2nd International Conference on Computer & Automation Engineering, vol. 2, pp. 65–67, 2010.

[14] Z. Chao-jie, H. Guo, and L. Shu-hai, "Test point selection of analog circuits based on fuzzy theory and ant colony algorithm," in Proceeding of IEEE AUTOTESTCON, pp. 164–168, 2008.

[15] Z. Wang, C. Guo, Q. jiang, and Y. Cao, "A fault diagnosis method for transformer integrating rough set with fuzzy rules,"Transaction of the Institute of Measurement and Control, vol. 3, pp. 243–251, 2006.

[16] J.-L. Chen and N. Rao, "A fuzzy expert system for fault diagnosis in electric distribution systems," inProceeding of Canadian Conference on Electrical and Computer Engineering, vol. 2, pp. 1283–1286, 1993.

Design and Implementation of Rough Set Algorithms on FPGA: A Survey

Kanchan Shailendra Tiwari
Research Scholar, ECE Dept. VNIT, Nagpur
Asst. Professor, E&TC Dept. MESCOE,
Pune, India

Ashwin. G. Kothari
Associate Professor, ECE Dept.
VNIT
Nagpur, India

***Abstract*—Rough set theory, developed by Z. Pawlak, is a powerful soft computing tool for extracting meaningful patterns from vague, imprecise, inconsistent and large chunk of data. It classifies the given knowledge base approximately into suitable decision classes by removing irrelevant and redundant data using attribute reduction algorithm. Conventional Rough set information processing like discovering data dependencies, data reduction, and approximate set classification involves the use of software running on general purpose processor. Since last decade, researchers have started exploring the feasibility of these algorithms on FPGA. The algorithms implemented on a conventional processor using any standard software routine offers high flexibility but the performance deteriorates while handling larger real time databases. With the tremendous growth in FPGA, a new area of research has boomed up. FPGA offers a promising solution in terms of speed, power and cost and researchers have proved the benefits of mapping rough set algorithms on FGPA. In this paper, a survey on hardware implementation of rough set algorithms by various researchers is elaborated.**

Keywords—Rough set theory; Discernibility matrix; reduct; Core; FPGA; classification

I. INTRODUCTION

Rough set theory(RST), by Zdzisław Pawlak, is a powerful mathematical tool, for discovering data dependencies by reducing the number of attributes contained in a data set using the data alone, without requiring any further additional information like degree of membership, probability etc. as required in fuzzy or in probability theory[1]. It is not an alternative to classical set theory but rather embedded in it. It provides efficient algorithms for finding hidden patterns in data, minimal sets of data (data reduction), evaluating significance of data, and generating sets of decision rules from data. The rough set approach is easy to understand, offers straightforward interpretation of obtained results, most of its algorithms are particularly suited for parallel processing. It is considered as one of the first non-statistical approach in data analysis [2]. Its methodology is concerned with the classification and analysis of imprecise, uncertain, vague or incomplete information and knowledge. The conceptual foundation of rough set data analysis is the consideration that all perception is subject to granularity and the ability to classify is at the root of human intelligence [3].

RST has been widely used in machine learning, data mining, and artificial intelligence successfully. Various software tools like ROSE, RSES, and ROSETTA [4-6] etc. are used for generating reduct, cores, and meaningful rules. These purely software program offer users a relatively high level of versatility and can handle any type of algorithm but the biggest and important issue is deterioration in the performance as the size of datasets increases. The software execution time becomes relatively slow while handling large real time datasets since the processor is not specially optimized for it. With the advent of digital technologies, Internet of Things, social media, etc. online storage of data has increased exponentially. It's need of the hour to process data in real time and at a faster rate. Recently, there has been a growing interest amongst researchers in developing a dedicated hardware for RST using FPGAs. The advantage of using a dedicated hardware is huge acceleration in terms of speed as they relieve main processor from the computational overheads. There are several such accelerators already available commercially in markets like Graphics Processing Units (GPUs), Digital Signal Processor (DSP), Fuzzy Processor. A dedicated hardware of rough set modules tends to be much faster than their software counterpart. The growth of VLSI industries had led to significant improvement in FPGAs in terms of resources available, speed, cost, and re-programmability etc. motivating researchers to choose it as one of the most viable solution.

In this paper in section 2, the basics of rough set theory are presented. In Section 3, need for hardware accelerator is discussed while section 4 covers the current status of art in the design of Rough Set Processor (RSP) by various authors followed by conclusion in section 5.

II. ROUGH SET PRELIMINARIES

The information in the world surrounding us is often imprecise, incomplete and uncertain. The human's ability of thinking and concluding widely depends on this information. In order to draw conclusion, one has to process this incomplete and imprecise data [7].

A soft computing tool mimics human decision making system and hence gives more promising results while handling such data. The various soft computing tools are fuzzy theory, neural network, genetic algorithms, rough set theory, etc. Rough set and fuzzy set theory are complementary to each other. RST is an effective tool for mining deterministic rules from a database. The rough set philosophy is founded on the assumption that with every object of the universe of discourse we associate some information i.e., knowledge is associated, through which classification can be achieved. It is based on

the idea that lowering the degree of precision in the data makes data pattern more perceptible [7]. The main motto of Rough Set theory is "Let the Data Speak for themselves". RST gives more formal framework for discovering facts from imperfect data. It gives results in the form of classification or decision rules derived from a set of examples.

Objects characterized by the same information are indiscernible (similar) in view of the available information about them. The indiscernibility relation generated in this way is the mathematical basis of rough set theory. Any set of all indiscernible (similar) objects is called an elementary set (neighborhood), and forms a basic granule (atom) of knowledge about the universe (fig.1). Any union of elementary sets is referred to as crisp (precise) set - otherwise the set is rough (imprecise, vague). Some of the Rough set related terms are presented below [7][8]:

A. Information System

The basic vehicle for data representation in the rough set framework is an information system (IS). An IS is a table, listing attributes of objects. Each row represents objects while each column specifies its attributes or features. Formally IS can be defined as IS = (U, A) where U is finite set of objects, $U=\{x_1, x_2, x_3, \ldots, x_n\}$; and A is a finite set of attributes (features, variables), the attributes in A are further classified as condition attributes C and decision attribute D, such that $A=C \cup D$ and $C \cap D=\emptyset$ (empty). Table 1 shows an example of a typical information system.

TABLE I. AN INFORMATION SYSTEM

Objects.	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	d
x_1	1	1	0	0	1	1	0	0	1
x_2	0	0	1	1	1	1	1	0	2
x_3	1	0	1	1	1	1	0	0	3
x_4	1	0	0	0	1	1	1	1	4
x_5	1	1	1	1	0	0	0	1	2
x_6	1	0	1	0	0	1	1	1	3
x_7	1	1	1	0	0	0	1	1	4
x_8	0	0	0	1	0	1	0	0	1

B. Decision Attributes

These are those attributes, which absolutely decide to which class the object belongs. In an IS shown in Table 1, d column is decision attribute column. The value of d, in it ranges from 1 through 4. Hence above IS is a 4 class system.

C. Condition Attributes

These are those attributes which do not absolutely decide the class to which the object belongs, but helps to decide. IS with distinguished decision and condition attributes are called decision tables. In Table 1, c_1, c_2, c_3---c_8 are condition attributes of 8 objects.

D. Upper Approximation ($\bar{A}(x)$)

Upper Approximation is a description of the objects that possibly belong to the subset of interest.

E. Lower Approximation ($\underline{A}(x)$)

It consists of those objects that can be with certainty classified as belonging to X. It is also known as POS(X).

F. Boundary Region

A set is said to be rough if its boundary region is non-empty, otherwise the set is crisp. It is also known as BR(X) Whereas U - $\bar{A}(x)$ is known as NEG(X). If the boundary region is a set $X = \emptyset$ (empty), then the set is considered "Crisp", otherwise, if the boundary region is a set $X \neq \emptyset$ the set X "rough" is considered.

G. Indiscernibility relation

Indiscernibility relation is a central concept in RST and is considered as a relation between two objects or more, where all the values are identical in relation to a subset of considered attributes. Indiscernibility relation is an equivalence relation, where all identical objects of set are considered as elementary.

H. Discernibility Matrix

An information system can also be presented in terms of a discernibility matrix. A discernibility matrix is a square matrix in which rows and columns are objects, and cells are attribute sets that discern objects. Two objects are considered discernible if and only if they have different values for at least one attribute. The discernibility matrix, denoted by M, for a decision table DT, of an IS is given as –

$$c_{ij} = \begin{cases} \Phi; & f_D(x_i) = f_D(x_j) \\ a \in A; & a(x_i) \neq a(x_j), f_D(x_i) \neq f_D(x_j) \end{cases} \quad (1)$$

A discernibility function can be constructed from discernibility matrix by OR-ing all attributes in c_{ij} and then AND-ing all of them together. After simplifying the discernibility function using absorption law, the set of all prime implicants determines the set of all reducts of the information system. However, simplifying discernibility function for reducts is a NP-hard problem. In Table 2 partial discernibility matrix for IS shown in Table 1 is tabulated.

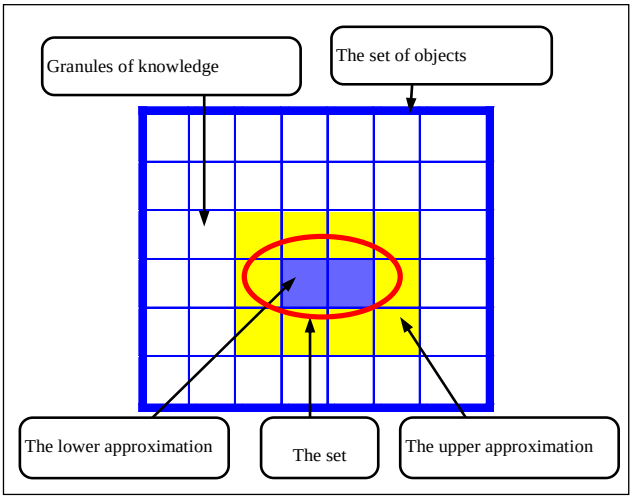


Fig. 1. Rough Set Concept Illusttration.

TABLE II. PARTIAL BINARY DISCERNIBILITY MATRIX

Objects.	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
x_{12}	1	1	1	1	0	0	1	0
x_{13}	0	1	1	1	0	0	0	0
x_{14}	0	1	0	0	0	0	1	1
x_{15}	0	0	1	1	1	1	0	1
x_{16}	0	1	1	0	1	0	1	1
x_{17}	0	0	1	0	1	1	1	1

I. Reduct and Core

The reduct and the core are important concepts in rough sets theory. A reduct is any minimal subset of condition features, which discerns all pairs with different decision values and is complete if the deletion of any attribute of a reduct will make at least one pair of objects with different decision attribute values indiscernible. The intersection of all reducts is called the core of the decision table. Discernibility matrix and Positive region based methods are more popular for computation of reducts in RST. Reducts can be of dynamic types too. Dynamic reducts are just a subset of all reducts which are derivable both from the original decision table and from the majority of randomly chosen decision sub-tables. Dynamic reducts gives dynamic rules.

J. Inconsistent Decision Table:

A decision table is inconsistent if for a given pair of object, all condition attributes are same but differ in decision attribute i.e. belong to two different classes. A medical database of 6 patients having symptoms of flu is shown in Table 3. The symptoms of flu are conditions attributes, which includes headache, muscle-pain, and temperature etc. while whether the patient is suffering from flu or not (1 or 0) is indicated by last column, also called as decision attribute. In Table 3, object 2 and 5 makes database inconsistent.

TABLE III. INCONSISTENT DECISION TABLE

Patients	Attributes			Decision
	Headache	***Muscle-pain***	***Temperature***	***Flu***
P_1	No	Yes	High	No
P_2	Yes	No	High	Yes
P_3	Yes	Yes	Very High	Yes
P_4	No	Yes	Normal	No
P_5	Yes	No	High	No
P_6	No	Yes	Very High	Yes

III. NEED OF HARDWARE ACCELERATORS

In data mining, processing of large volumes of data using complex algorithms is increasingly common. There are numerous applications like image processing, speech processing, artificial intelligence, analyzing experimental data etc. which demands fast processing of high volumes of data.

Computers are able to handle a wide variety of applications. Since the design and development of computers from 1940, there has been exponential growth in its performance for decades. This growth has been further complemented by a combination of improvements in implementation technology, architectural innovations, and compiler optimizations. However, as computers becomes even faster, new applications empowered by technology arise, which demands development of new technologies [9]. In addition to those continuous improvements, designers have relied on solutions based on special architectures to accelerate the performance of these applications, with processing units exploiting their common features such as parallelism, repetitive tasks or intensive mathematical processing. Traditionally, these solutions have been of two types:

A. Parallel Processing Computers With Parallel Processors

During the last few decades, traditional general-purpose single-core CPUs has shown a remarkable growth due to the multiple improvements in VLSI technologies. This growth was marked by the reduction in size of transistors, increase in the frequency of processor as per Moore's law and hence software performance also improved continuously for decades. However, the gain in the performance of conventional single core CPU has diminished as the VLSI system performance hit the memory wall, power wall [10] and instruction-level parallelism (ILP) wall. The memory wall refers to the increasing gap between processor and memory speeds. This demanded increase in size of cache for hiding memory access latencies [11]; thus making memory bandwidth a bottleneck in performance. The power wall refers to power supply limitations and thermal dissipation limitation. For the silicon lithography below 90nm, the static power from leakage current surpass dynamic power from circuit switching. Power density has become the dominant constraint in chip design, and limits the clock frequency growth [12]. The performance, cost, and reliability of modern computer systems and data centers are dictated by the management of their limited energy and thermal budgets [13]. The ILP wall refers to the rising difficulty in finding enough parallelism in the existing instructions stream of a single process. Increasing cache size or introducing more ILP yields too little performance gain compared to the development cost [14]. Together, these three walls reduce the performance gains expected for single-core general-purpose processors.

With current technology, even though the number of transistors is increasing, but the clock speeds are flattening as shown in fig.2. In order to overcome the problems posed by power and ILP wall, the computer industry shifted from single core processors to multiple parallel processing units. This showed the beginning of a paradigm shift towards parallel hardware architectures. CPU manufacturers used the improved processes to fit more and more CPU cores onto each device, producing generations of many-core processors, each running at about the same clock frequency as their predecessors.

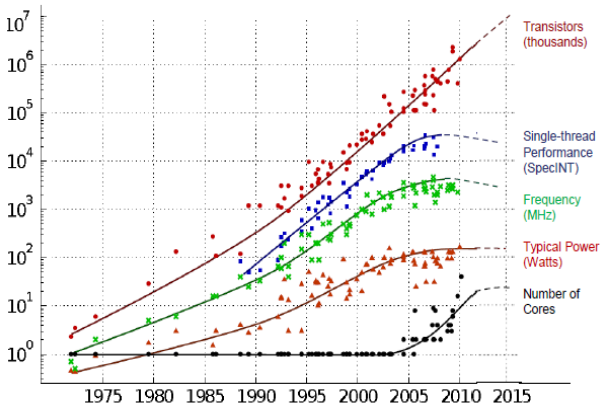


Fig. 2. Moore's Law- The number of transistors and power consumption is constantly increasing, while the frequency is flattening. (Source- Taken from Kunle Olukotun and Herb Sutter)

However, conventional computer programs are described as sequentially executed instructions and cannot easily be adapted for a multi-processor environment. This condition prevents a potential speed-up due to the problem of finding enough parallelism in the software. The speedup S from using N parallel processing units is calculated using Amdahl's law [15] as

$$S = \frac{1}{(1-p)+\frac{p}{N}} \qquad (2)$$

where p is the fraction of the sequential program that can be parallelized.

If one assumes that 50% of the sequential code can be executed on parallel processors, then speedup will still be limited to a modest factor of 2, no matter how many parallel processors are used. Parallel architectures have a promising future, but will require new design approaches and programming methodologies to enable high system utilization. This means that for faster execution, one must actively seek alternative ways to speed up the software.

B. Accelerators

In order to exploit the parallelism and distribute the computation amongst several processing cores, software execution style should change from sequential to parallel. This opens up the playfield for new types of processing resources to complement the traditional CPU architecture. Recently, market is dominated by cost-efficient accelerators available from several vendors as common off the-shelf (COTS) products [16]. Accelerators are specialized processors that can be used to speed up specific processing tasks and they complement conventional architectures. Accelerators with CPUs, forms a hybrid computing system or Multi-Processor Systems-on-Chip (MPSoCs), where each processing resource executes the parts of the software for which it delivers the best performance. Currently, heterogeneous MPSoCs are becoming the de-facto standard for embedded system design. Such system usually is composed of several general purpose processors, digital signal processor and hardware accelerators interconnected through various communication mechanisms for accelerating specific part of an application. This results in greatly increased system performance.

The main competitors for the COTS accelerator market are Field Programmable Gate Arrays (FPGAs) and Graphics Processors (GPUs). These devices have strong mass markets in the high performance computing fields. Acceleration continues to be a great necessity in this new scenario dominated by multicore processors and clusters built with them, because of the following reasons:

- Optimum performance for all types of applications is not given by General Purpose Processors, even if multicore technology is used.
- There are certain applications like single thread applications, embedded systems, etc., where significant acceleration is not achieved by using conventional multicore technology.
- The complexity and huge size of digital circuit causes the run time of software to become unreasonably large as these problems are NP-hard.
- To reduce execution time.
- To offload the general purpose CPU.
- To offer special features for easy use.

Therefore, while the use of specific parallel processors computing has declined, new solutions continue to appear in the field of hardware accelerators. Accelerators can be realized using different technologies like DSP, GPGPU, ASIC, FPGA.

They all differ in architectures and are suited for different applications.

1) Digital Signal Processor (DSP)

- A DSP is a processor system optimized to implement signal processing at very high speed.
- DSP's include a specialized architecture which allows parallel processing at the instruction level; this is called SIMD (Single Instruction Multiple Data).
- There are fixed and floating point DSP's available in market.
- Parallel instructions are used with special assembler instructions included in C program.
- The DSP Blackfin 609 is a fixed point DSP based in a Dual-Core processor working up to 1GHz. The Blackfin arithmetic unit allows the execution of multiple operations in parallel: up to four 8-bit video ALUs or two multiplications and 2 accumulations of 32/40-bits.

2) Application Specific Integrated Circuit (ASICs).

- ASIC is basically a circuit designed for a specific use rather than a circuit designed for general purposes
- ASIC designs offer a very attractive solution for many high volume applications.

- The design using ASIC offers better performance, density and power consumption when compared to an FPGA.
- ASIC prototyping can be done using FPGAs, which allows taking advantage of FPGAs re-programmability.
- However, the cost of prototyping is quite high increasing the Nonrecurring Engineering (NRE) costs depending upon the design, complexity and method of implementation.
- Also, they do not offer any flexibility, as the task they perform cannot be modified.
- Hence, their use for acceleration purpose is quite limited.

3) General Purpose Graphical Processing Units (GPGPUs).

- Graphics Processors are highly parallel processors capable of running thousands of threads simultaneously. Threading is handled automatically by the hardware thread manager. The programmer does not have direct control of the processors of the GPU; everything is done through Application Programming Interfaces (API).
- They are special types of processor dedicated for graphics operation in game consoles and computers.
- They are an order of magnitude faster on floating point operations.
- Recently GPGPUs have been specifically developed with the computational precision required for finite element analysis solutions as well as the computational power to effectively complement the performance of the latest CPUs.
- With hundreds of low-power cores on a single socket, they have the potential to dramatically increase computing capacity, provided that the compute workload will fit in the available memory of the GPGPUs.
- However, the applications with complex feedback loop, and control or extensive bit handling is not suitable for GPGPUs implementation. The high power consumption of GPGPUs restricts their usage to certain applications.
- GPGPUs are difficult to program for general-purpose uses.
- In the current market there are three principal GPU providers: NVidia, Intel, and AMD.

4) Field Programmable Gate Arrays (FPGAs)

- FPGA is semiconductor device, invented by Xilinx co-founder, Ross Freeman, in 1984.
- FPGAs generally consist of sets of flexible gates, registers, and memories whose function and interconnection are controlled through the loading of SRAMs (Static Random Access Memory).
- FPGA can be programmed either statically (between applications) or dynamically (during an application) without the addition of physical hardware elements.
- It is intended to fill the gap between the hardware (ASICs) and software (General Purpose (GP) Processors), achieving potentially much higher performance than software, while maintaining a higher level of flexibility than hardware
- The main resources available in the current FPGAs are hard Processors, RAM memory, Slices, DSP Slices, Multipliers, Gigabit transceivers, Triple-Speed Ethernet MAC, PCI express, Phase Locked Loop (PLL), etc.
- FPGAs tend to operate at relatively modest clock rates measured in a few hundreds of MHz, but they can perform sometimes tens of thousands of calculations per clock cycle while operating in the low "tens of watts" range of power.
- Improvements in FPGAs have driven a huge increase in their use in space, weight and power (SWaP) constrained embedded computing systems for military and aerospace applications. They are ideal for addressing many classes of military applications, such as Radar, SIGINT, image processing and signal processing where high-performance DSP and other vector or matrix processing is required.
- FPGAs seem to gives an unbeatable edge over a microprocessor as they can provide 50 to 100 times the performance per watt of power consumed than a microprocessor.
- FPGA offers field reprogrammability. A new bit stream file can be uploaded remotely.

The advantages and disadvantages of FPGA with respect to other available technologies are presented in table 4. In case of the ASIC, the fabrication cost is reduced if chip is produced in mass; however for unit production ASIC design is expensive. FPGA combines many benefits of both software and ASIC implementations. Like software, the mapped circuit is flexible, and can be reconfigured over the lifetime of the system. FPGAs therefore have the potential to achieve far greater performance than software as a result of bypassing the fetch-decode-execute operations of traditional processors, and possibly exploiting a greater level of parallelism. Creating parallel programs implemented in FPGAs is not trivial. Fig. 3 [17] summarizes the application fitness categorization. Hence it is concluded that FPGA is best choice for implementation of rough set algorithms as it out performs with respect to other available technologies.

	GPGPU	FPGA
Good to Fit ↕ Bad Fit	No inter-dependences in the data flow and the computation can be done in parallel	Computation involves a lot of detailed low level hardware control operation which cannot be efficiently implemented in high level languages, such as bit operation
	Applications contain a lot of parallelism, but involve computations which cannot be efficiently implemented on GPU	A certain degree of complexity is required, and the implementation can take advantage of data streaming and pipelining
	Applications have a lot of memory access and have limited parallelism	Applications that require a lot of complexity in the logic and data flow design

Fig. 3. Application Characteristic Fitness Categorization

TABLE IV. TECHNOLOGY COMPARISON

Features	Platforms				
	DSP	*GPU*	*ASIC*	*FPGA*	*GPP*
Size	Medium	-	High	High	Medium
Power	Medium	-	High	Low	Medium
Flexibility	High	High	-	High	High
Reliability	Low	Low	High	Medium	Low
Parallelism	Low	Medium	High	High	Low
Operation Frequency	Medium	High	High	Low-Medium	High
Design complexity	Medium	Low	-	-	High
Cost	Medium	Low	High	Low	High

IV. CURRENT STATE OF ART

A. Z. Pawlak Concept of Rough Set Processor

The concept of Rough Set Processor (RSP) is put forth by Z. Pawlak (RSP) in [18]. He stated that RSP can be used as an additional fast classification unit in ordinary computers or as an autonomous learning machine. In latter case, RSP can replace neural networks. He stated that each row of a decision table induces a rule, which specifies the actions if some conditions are satisfied. If a decision rule uniquely determines a decision in terms of condition attributes then that rule is certain otherwise it is uncertain. According to him, decision rules are closely associated with concept of approximation in rough set theory. Lower approximation are described by *certain* decision rules while upper approximation by *uncertain* decision rules. He associated the two conditional probabilities called, *uncertainty* and *coverage coefficient* with each decision rule. The *certainty coefficient* expresses the probability that an object belongs to the decision class specified by the decision rule, if it satisfies the condition of the rules. The *coverage coefficient* gives the conditional probability of the reasons for a given decision. He proved that the *certainty* and *coverage* coefficient satisfy Bayes' theorem and it can be used for drawing conclusion from data. This idea is used as a foundation for RSP. The computation of certainty and coverage factors of decision rules is dependent on strength of decision rules. The strength can be computed from data or can be a subjective assessment. The concept of flow graph i.e. a directed acyclic graph is associated with decision table. In that graph, to every decision rule, a directed branch connecting input node with output node is assigned. The strength of the decision rule represents a throughflow of the corresponding branch. The classification of objects is done by finding the maximal output flow in the flow graph whereas, the explanation of the decisions is connected to the maximal input flow associated with the given decision. He proposed requirement of a special microprocessor for doing all above mentioned computation. According to him, RSP should perform operations pointed out by the flow graph of a decision table i.e. first computation of strengths from the support of decision rules, and then certainty and coverage factors of all rules should be computed. All these parameters are stored and computed subsequently in a format of word structure as shown in Fig. 4. Decision table will store condition and decision attributes of objects, Decision rule register will compute meaningful rules from data while arithmetic block will perform arithmetic operation of computing strength, coverage and certainty factors as shown in Fig. 5. This idea, however, is not realized on programmable logic devices.

Condition attribute	Decision attribute	Support	Strength	Certainty	Coverage

Fig. 4. Word structure.

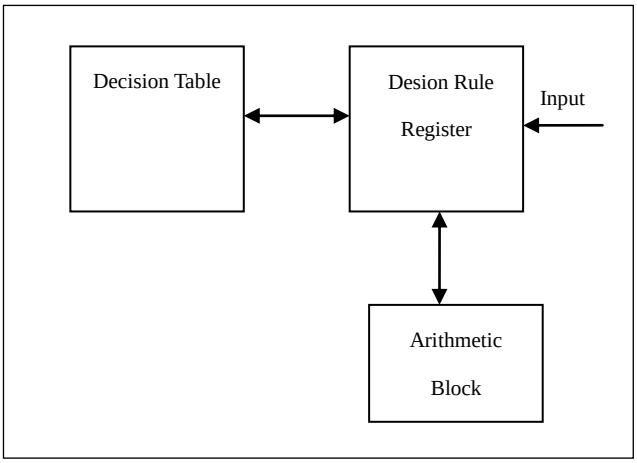


Fig. 5. RSP Structure.

B. Towards PRSComp

Authors in [19] put forward the concept of describing rough set methods using cellular networks. They designed a

device for parallel processing of rough set algorithms and called it as Parallel Rough Set Computer (PRSComp). Cellular network is a matrix of interconnected elements of the same type, wherein each cell is treated as a single processor and a set of control registers. The cellular network based on rough sets transforms the input data set to the matrix and performs basics operation of rough sets using matrix notations. In [19] authors have used all basic notions of rough set (indiscernibility relation, upper and lower approximation, reducts and core calculations) for implementation of PRSComp. Authors have given pseudo code for all basic routines also.

C. Lewis T. Idea as Learning Machine

Authors in [20] built a Universal Logic Machine (ULM) based on the principles of constructive induction and RST. It is a self-learning rough set model based on the concept of cellular networks by [19]. It is thought of an early prototype of data mining machine which will not only be able to collect data from online databases, but also from industries, military and other real time applications. The authors presented a preliminary work on design and implementation of a single instruction multiple data (SIMD) computer to implement RST operations. RST can be effectively used in logic minimization and data mining. They identified that some subsets of RST are isomorphic with some subsets of logic synthesis and decomposition theories; hence their mutual relationship can be investigated, leading to synergies of concepts. For example the powerful logic concepts of rough set theory can be linked with efficient algorithms and data structures developed in logic synthesis for EDA. According to them the RST algorithms have a natural high parallelism and high possible speed-ups. Using a fast prototyping tool, the DEC-PERLE-1 board based on an array of Xilinx FPGAs, a virtual SIMD processor that accelerates the learning (design) of optimized multi-valued logic nets using the concept of cellular networks has been developed. They have proposed the principles of learning hardware that will use previous human problem-solving experience and apply mathematical algorithms, problem-solving strategies rather than relying only on neural network and genetic algorithm.

A solution to a given problem is achieved by partitioning it in two phases: the phase of learning and the phase of using the knowledge. The hardware processor (parallel rough set computer) is responsible for creation of logic network description using logic or mathematical algorithms. The optimally constructed network is mapped on FPGA using EDA tool. The knowledge of machine is stored in memory. While solving the new problem under the supervision of software program in the main processor, the hardware switches between various learned nets, depending on rules. Since network has to solve new problems, hence new datasets and training decisions are accumulated and the network is repetitively automatically redesigned. The old network can serve as a platform for redesigning of new network or new network can start from scratch to avoid any bias.

Lewis implemented basic rough set operation of basic category, upper approximation, and lower approximation, indispensable and external comparison. Authors demonstrated the working of all above mentioned algorithms on a learning machine called as parallel rough set computer (PRSComp) and its architecture is shown above in Fig. 6. In Fig. 6 E is word selection register, C is comparand register and CM is column mask register. He proposed a machine consisting of m by n primitive processor, in which each of the processors is connected to its neighboring four processors as well as to global control signals. Each processor performs the same operation defined by the instruction at that time. PRSComp operates as SIMD (single input multiple data). The input data is mapped on these processors as a binary matrix of size m*n wherein each processor operates on one bit of it at a time. They utilized various registers for doing all these operations. In this paper, there is no discussion on time complexity, space complexity. The author however has put forth the problems posed by purely genetic and artificial neural network and justified that rough set theory is an appropriate solution for handling those problems.

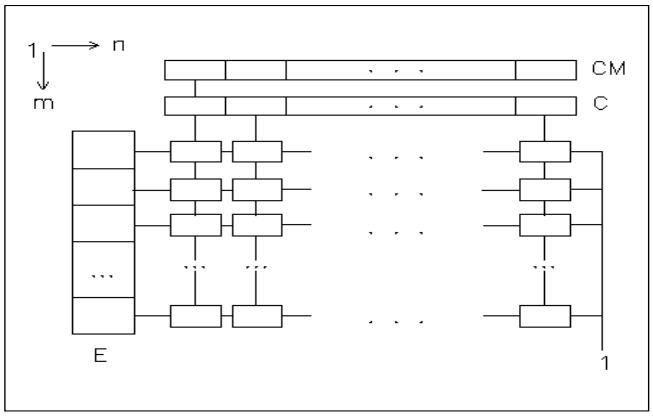


Fig. 6. PRScomputer architecture.

D. Kanasugi Discernibility Matrix approach

Authors in [21] presented a design of architecture of rough set processor in 2001 (shown in fig.7). It is used for solving large-scale problem in real time. The main blocks in their architecture are discernibility matrix maker, core selector, covering unit, reconstruction unit, registers, cache memory, controller and bus interface. The execution process is divided into two parts: pre-process and main process. In pre-process, some sparse terms are selected as cores and then implying relation reduces the input logic functions. In the main process, input logic function is converted into the sum of products form.

The block of discernibility matrix maker is not dealt in this proposed work. The core selector unit selects data whose sum is minimum and transfers its row number to core number register unit. Core unit reduces data using implying function. Reconstruction unit searches for dominant variables from input logic function and then reconstructs the important rules from it. Memory interface is identified as a potential bottleneck in the design. The work of [22] is an extension of [21]. In [21], only design has been proposed whereas in [22], synthesis, simulation and implementation on SPARTAN 3E board of Xilinx is presented. They minimized the discernibility matrix by obtaining a reduced discernibility function. The outputs of system are small logical functions

representing important decision rules. Authors have developed a co- processor, which will be interacting with memory for data retrieval and storage purpose. The system depends on external data source for creation of large logical functions from data base for correct operation and algorithm is based on approximation technique. Their co-processor is capable of dealing with objects of size 1000,000 and 2032 attributes. They have dealt with binary attributes, leaving discretization process, a task for future development. They have shown that their proposed processor is ten times faster than PC, though the clock frequency is about 70 times slower. There is no discussion on time complexity and space complexity. However, their algorithm is based on computing discernibility matrix and discernibility function, whose time complexity will no longer be less than O ($|U|^2|A|^2$).

E. G. Sun's FPGA implementation of RST

G. Sun in his paper [23] has implemented Rough set theory algorithms on FPGA in 2011. Author has provided a new and effective method for hardware fault diagnosis and verified the effectiveness of method through simulation. He has made use of genetic algorithm along with rough set theory and presented a case study of nonlinear aircraft model. He implemented discretization block, based on dependency degree. The breakpoints are deleted based on dependency degree. The reducts are calculated using genetic algorithm. The simulation results using Modelsim for discretization and attribute reduction has been presented. The algorithms are not purely based on RST; rather it is hybridization of rough set with genetic algorithms.

F. Maciej Kopczynski ´s et al. computation of reduct and core on FPGA

Maciej Kopczynski et al. in their paper [24 - 26] presented reduct and core generation algorithm based on discernibility matrix. They have presented hardware solution architecture for binary decision table. They have discussed architecture of discernibility and reduct block. They used VHDL simulator and the development board equipped with an Altera FPGA during the research. The reduct generation algorithm is simple and based on attribute count frequency [25]. The algorithm gives super reduct, however it does not discusses the case of breaking tie between two attributes having the same count value. They have also compared the time required for execution of reduct and core generation on software and hardware for varying size of database. They have randomly generated the binary database. Their results show a significant increase in the speed of data processing. In [26], they have shown three variants of discernibility matrix implementation. Authors have shown time required for computing reducts and cores for all three methods. The issue of dealing with larger databases is not handled.

G. K.S.Tiwari's et.al. Hardware Implementation

Tiwari et.al in their work [27] presented architecture for computing reduct using binary discernibility matrix. They have used Xilinx software and Spartan 3 FPGA. They have proposed a Rough Set Machine which generates rules for classification applications. The classification task concentrates on predicting the value of the decision class for an object among a predefined set of classes' values. This rough set machine uses the concept of discernibility matrix for calculating the reduct, and using these reduct it generates the rules which are used for classifying the objects. The Reduct block is synthesized and downloaded on FPGA in [28]. The architecture of binary discernibility matrix is shown in fig.8. In [29]; Quick reduct algorithm is used for computation of reduct for a medical database.

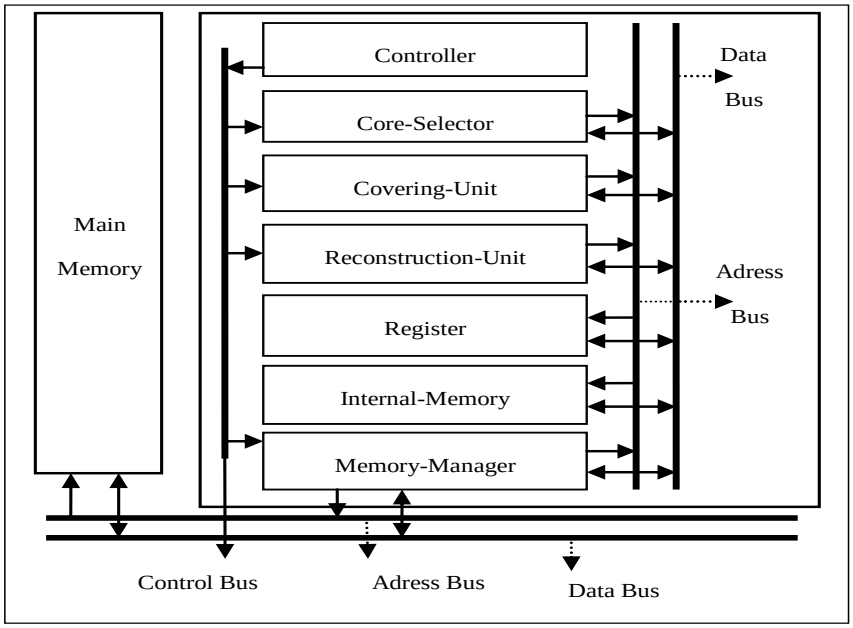


Fig. 7. Kanasugi's Proposed Block Diagram of Rough Set Processor.

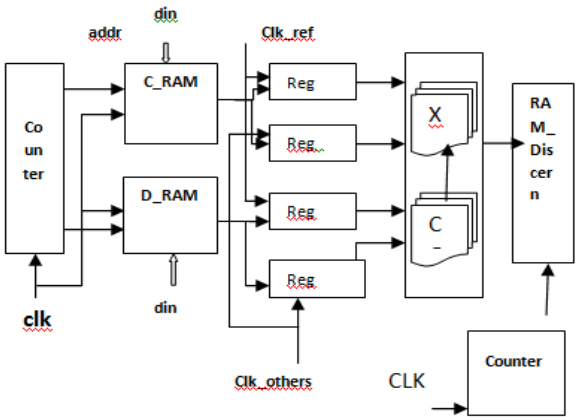


Fig. 8. Binary discernibility Matrix

TABLE V. SUMMARY

Sr.No.	Authors	Year	Brief Summary
1	Mieczyslaw Muraszkiewicz, and Henryk Rybinski	1994	Concept is based on Indiscernibility relation, Lower and upper approximation
2	Lewis et.al	1999	Self-learning hardware model based on cellular concept, Implementation done Xilinx board.
3	Kanasugi	2001	Algorithm is based on Discernibility matrix
4	Z.Pawlak	2004	Decision Flow graph used for representing tables.
5	Kanasugi and Mitsuhiro Matsumoto	2007	Discernibility matrix based algorithm proposed and implemented on Spartan 3E Board.
6	G Sun et. al	2011	Genetic based attribute reduction system; discretization is based on dependency approach of RST.
7	K.S.Tiwari et.al	2011	Concept of discernibility matrix used for generation of reducts and rules.
		2012	Pipelining and use of Dual port RAM as a part of extension.
		2013	Quick Reduct algorithm based on dependency function is implemented and simulated using ISIM.
8	Maciej Kopczynski et. al	2013	Computation of short reduct and core based on discernibility matrix. Huge acceleration achieved.
		2014	Discernibility matrix built using three different methods

V. CONCLUSION

In this paper a survey on hardware implementations of Rough set algorithm is presented. It is summarized in brief in table 5. A lot of research work is carried out on rough set theory using software; however hardware implementation is still not much explored. With exponential growth in quantity of data collected, its need of hour to process data fast, and extract meaningful rules from it. FPGA offers a promising solution to deal with such kind of problems as rough set algorithms are inherently parallel. Thus these algorithms can be effectively mapped on FPGA.

REFERENCES

[1] Zdzisław Pawlak, Andrzej Skowron,"Rudiments of rough sets," Information Sciences, vol. 177, January 2007, pp. 3-27.

[2] Zdzisław Pawlak , "Rough Sets," International Journal of Computer and Information Sciences, vol.11, September 1982, pp. 341–356.

[3] Zdzisław Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, Boston, London, 1991.

[4] Bart lomiej Predki et al, " ROSE - Software Implementation of the Rough Set Theory ," Rough Sets and Current Trends in Computing, L. Polkowski and A. Skowron ,Eds., LNCS 1424, Springer-Verlag Berlin Heidelberg 1998, pp. 605-608

[5] Andrzej Skowron et al. Logic Group, Institute of Mathematics, Warsaw University, Poland,1994, Accessed Aug. 2010. http://logic.mimuw.edu.pl/~rses/.

[6] M Kierczak et al., ROSETTA Development Team, 2009, Accessed Aug. 2010. http://www.lcb.uu.se/tools/rosetta/.

[7] B. Walczak, D.L. Massart, "Rough sets theory", Chemometrics and Intelligent Laboratory Systems, vol. 47/1, April,1999, pp. 1–16.

[8] Silvia Rissino, Germano Lambert-Torres "Rough Set Theory – Fundamental Concepts,Principals, Data Extraction, and Applications," *Data Mining and Knowledge Discovery in Real Life Applications*, Julio Ponce , Adem Karahoca,Eds, I-Tech Education and Publishing,2009

[9] Altera. Accelerating high-performance computing with fpgas. white paper, Altera, October 2007.

[10] K. Gulati and S. P. Khatri. Hardware Acceleration of EDA Algorithms: Custom ICs, FPGAs and GPUs. Springer, 2010

[11] D. A. Patterson. Latency lags bandwith. Commun. ACM, 47(10):71–75, 2004

[12] J. Shalf. The new landscape of parallel computer architecture. journal of Physics, 78, 2007.

[13] J. Carter and K. Rajamani. Designing energy-efficient servers and data centers. IEEE Computer, 43(7):76–78, 2010

[14] A. R. Brodtkorb, C. Dyken, T. R. Hagen, J. M. Hjelmervik, and O. O. Storaasli. State-of-the-art in heterogeneous computing. Scientific Programming, pages 1–33, 2010.

[15] Amdahl, Gene M. "Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities, Reprinted from the AFIPS Conference Proceedings, Vol. 30 (Atlantic City, NJ, Apr. 18–20), AFIPS Press, Reston, Va., 1967, pp. 483–485, when Dr. Amdahl was at International Business Machines Corporation, Sunnyvale, California." Solid-State Circuits Society Newsletter, IEEE 12.3 (2007): 19-20.

[16] Mitrion, Low power hybrid computing for efficient software acceleration,White paper,Mitrionics,2008

[17] S. Che, J. Li, J. Sheaffer, K. Skadron, and J. Lach. Accelerating compute-intensive applications with gpus and fpgas. In Symposium on Application Specific Processors, pages 101 –107, 2008.

[18] Pawlak, Z, "Elementary Rough Set Granules: Toward a Rough Set Processor," Rough-Neural computing Cognitive Technologies, Dr.. S.K. Pal et al, Springer 2004, pp.5-13.

[19] Muraszkiewicz, Mieczyslaw, and Henryk Rybinski. "Towards a parallel rough sets computer." Rough Sets, Fuzzy Sets and Knowledge Discovery. Springer London, 1994. 434-443.

[20] Lewis, M. Perkowski, and L. Jozwiak, "Learning in Hardware: Architecture and Implementation of an FPGA – Based Rough set machine," IEEE, 1999, pp.326-334.

[21] A. Kanasugi, "A Design of architecture for Rough Set Processor", New Frontiers in Artificial Intelligence, T Terano et al. ,Ed.,LNCS, vol. 2253, 2001, pp.406-410.

[22] A.Kanasugi and M. Matsumoto, "Design and Implementation of Rough Rules Generation from Logical Rules on FPGA board", *Rough Sets and Intelligent Systems Paradigms*, M. Kryszkiewicz et al, Eds,LNCS, vol. 4585, 2007, pp. 594-602.

[23] Guoqiang Sun; Xiaoming Qi; Yuanyuan Zhang, "A FPGA-based implementation of Rough Set Theory," Control and Decision Conference (CCDC), 2011 Chinese, vol., no., pp.2561-2564.

[24] Stepaniuk, Jaroslaw, Maciej Kopczynski, and Tomasz Grzes. "The First Step Toward Processor for Rough Set Methods." Fundamenta Informaticae 127.1 (2013): 429-443.

[25] Grześ, Tomasz, Maciej Kopczyński, and Jarosław Stepaniuk. "FPGA in Rough Set Based Core and Reduct Computation." Rough Sets and Knowledge Technology. Springer Berlin Heidelberg, 2013. 263-270.

[26] Kopczynski, Maciej, Tomasz Grzes, and Jaroslaw Stepaniuk. "Generating Core in Rough Set Theory: Design and Implementation on FPGA." Rough Sets and Intelligent Systems Paradigms. Springer International Publishing, 2014. 209-216.

[27] Tiwari, K. S., and A. G. Kothari. "Architecture and Implementation of Attribute Reduction Algorithm Using Binary Discernibility Matrix." Computational Intelligence and Communication Networks (CICN), 2011 International Conference on. IEEE, 2011.

[28] Tiwari, Kanchan S., Ashwin G. Kothari, and Avinash G. Keskar. "Reduct generation from binary discernibility matrix: an hardware approach." International Journal of Future Computer and Communication 1.3 (2012): 270-272.

[29] Tiwari, Kanchan, Ashwin Kothari, and Riddhi Shah. "FPGA Implementation of a Reduct Generation Algorithm based on Rough Set Theory." International Journal of Advanced Electrical and Electronics Engineering ISSN (Print) : 2278-8948, Volume-2, Issue-6, 2013 [55-61]

Dynamic Programming Method Applied in Vietnamese Word Segmentation Based on Mutual Information among Syllables

Nguyen Thi Uyen
IT Faculty - Vinh University

Tran Xuan Sang
IT Faculty - Vinh University

***Abstract*—Vietnamese word segmentation is an important step in Vietnamese natural language processing such as text categorization, text summary, and automated machine translation. The problem with Vietnamese word segmentation is complicated because Vietnamese words are not always separated by a space. One word can include one or more syllables depending on the context. This paper proposes a method for Vietnamese word segmentation based on the mutual information among the syllables combined with dynamic programming. With this method, we can achieve an accuracy rate of about 90% with a raw text corpus.**

Keywords—Vietnamese word segmentation; dynamic programming; mutual information; Vietnamese syllables

I. INTRODUCTION

Word segmentation is the process to determine the boundaries between words in sentences. Words in the Vietnamese language are not always separated by blank spaces. A word may contain several syllables. The syllables are combined to form different words depending on the context of the text. Therefore, it is difficult to solve this problem automatically. Example 1: The sentence written in Vietnamese "Học sinh học sinh học" - in English "The pupils study biology". This sentence is composed of two Vietnamese syllables "học ~ study" and "sinh ~ biology" which form different words in the sentence. The correct solution should be "học sinh | học | sinh học" ~"The pupils | study | biology". One of the most difficult tasks in Vietnamese word segmentation is to determine the ambiguities of the sentence.

The same sentence may have different word segmentation solutions if it is in a different context. Example 2: The sentence written in Vietnamese "Ông già đi nhanh quá" may have two different meanings. One is "The old man goes too fast", the other one is "Grandfather gets old too fast". It results in two word segmentation solutions: Ông già| đi |nhanh quá and Ông| già đi| nhanh quá. In this case, it is needed to consider the context of this sentence in order to select the best solution.

Mutual information (MI) between the syllables presents the correlation of syllables to be combined as a word. The greater MI value will show the higher probability of words combination of syllables. The MI theory will be presented in more details in section 3.a.

The dynamic programming technique is used to reduce the complexity of the computation. This method will be presented in section 3.b.

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 describes the proposed method. Section 4 provides the experimental results. Finally, section 5 summarizes the work of this paper.

II. RELATED WORKS

This section presents the previous works in Vietnamese word segmentation.

A. Maximum Matching Method

Maximum matching algorithm is commonly used to word segmentation problem. The idea of this method is to start at the first syllable in a text and attempt to find the longest word starting with that syllabus in the dictionary. If a word is found, the maximum matching algorithm marks a boundary at the end of the longest word, then begins the same longest match search starting at the syllable following the match. Whereas, that syllable is segmented as a word, and begins the search starting at the next syllable [Wong et al, 1996] [1]. Dinh et al, 2001 used Maximum Matching method to segment Vietnamese word [2]. However, the accuracy of word segmentation is not high.

B. Transition Graph Method

In this method, each syllable is represented by a vertex. The edge represents weight of connection between two syllables which is calculated based on the data training process. The transition graph will show the probability among syllables to form the words in a specific text. Nguyen et al, 2003; Pham et al, 2009 used this method to segment Vietnamese word [3,4].

C. Support Vector Machine Method(SVM)

Point-wise machine learning method (SVM) is used to mark two kinds of symbols: space (word segment symbols) and underscore (linking two syllables symbol) (Luu et al, 2012) [5]. There are three basic features in point-wise methods: n-grams of syllables, n-gram types of syllables, and featured dictionary. Vietnamese language has about 70% of the words with 2 syllables, and 14% words with 3 syllables, therefore the point-wise window was set as w = 3. The author defines four types of syllables: uppercase syllables (U): Vietnamese syllables begin with capital letters. lower syllables (L): the Vietnamese syllable contains only lowercase letters. Numerical syllables (N) only consists of the digits. The other type (O): the syllables belongs to a foreign language. This research achieved 98.2% accuracy rate. However, this method needs a good featured dictionary.

D. Combination Method

Le et al, 2008 [6] combines a finite state machine, canonical form analysis, maximum matching method to segment Vietnamese word. Minimal finite-state automaton is used to present the Vietnamese lexicon. A text to be tokenized is first parsed into lexical phrases and other patterns using pre-defined regular expressions. The automaton is then deployed to build linear graphs corresponding to the phrases to be segmented. A tool named vnTokenizer is then created to show the effectiveness of this method.

III. PROPOSED METHOD

For Vietnamese, there is a lack of large lexicographic resources, and annotated corpora are also rare, therefore we will develop a method that only rely on raw corpus. The mutual information (MI) is a statistical score which helps to segment the words. The mutual information is calculated based on the frequency of the syllables in a raw corpus. The main ideal of our method is to maximize the MI-score of the chunk, using different segmentations. We will calculate all the possibilities of segmentations for a given sentence. Which possibility has the highest MI-score, becomes the final solution. There are some difficulties in the calculation. First, the segmentation possibility is an exponential function of the length of the sentence. A long sentence will cause a large number of ways of segmentation. Other problems consist of the difficulty of calculating MI-score, and sparse data. In order to overcome those difficulties, the dynamic programming is utilized. In following sub-section, MI-score and dynamic program applied in word segmentation are presented in detail.

A. Corpus and MI-Score

We build a corpus by collecting text from many Vietnamese websites and online news papers. Our raw corpus contains about 41 million syllables.

Mutual information is an important factor to identify the correlation between syllables in a corpus. The equation to calculate MI value is presented as below (Ong & Chen, 1999).

$$MI(cw) = \frac{p(cw)}{p(lw) + p(rw) - p(cw)} \quad (1)$$

Where:

- *cw* is a chunk containing n syllables. $\mathrm{cw} = \mathrm{c_1c_2 \dots c_n}$.

- *lw* is a chunk containing n-1 syllables $\mathrm{lw} = \mathrm{c_1c_2 \dots c_{n-1}}$.

- *rw* is a chunk containing n-1 syllables $\mathrm{rw} = \mathrm{c_2c_3 \dots c_n}$.

The higher *MI(cw)* value shows a higher probability of *lw* and *rw* appearing in the corpus. It means *cw* has high probability to be a compound word.

Based on equation 1, we elaborate the way to calculate MI-score for certain segmentation.

Given a sentence $\mathrm{C = c_1c_2 \dots c_n}$ with $\mathrm{c_i}$ is a syllable.

- *N*: total number of syllables in the corpus.
- f(w): frequency of chunk *w* in the corpus.
- p(w) :Probability of chunk *w* in the corpus.

$$p(w) = \frac{f(w) + 1}{N} \quad (2)$$

- $\mathrm{MI(c_1c_2)}$ *:* Mutual Information value of two syllables $\mathrm{c_1, c_2}$.

$$\mathrm{MI}(c_1c_2) = \frac{p(c_1c_2)}{p(c_1) + p(c_2) - p(c_1c_2)} \quad (3)$$

- $\mathrm{MI(c_1c_2 \dots c_n)}$: Mutual Information value of n syllables $\mathrm{c_1, c_2, \dots, c_n}$.

$$MI(c_1c_2 \dots c_n) = \frac{p(c_1c_2 \dots c_n)}{p(c_1c_2 \dots c_{n-1}) + p(c_2, c_3 \dots c_n) - p(c_1c_2 \dots c_n)} \quad (4)$$

Given a sentence with certain segmentation as belows.

$$t(a) = w_1|w_2| \dots |w_m$$

- Then, the MI-score of this segmentation is calculated as below.

$$\mu(t(a)) = MI(w_1) + MI(w_2) + \cdots + MI(w_m) \quad (5)$$

B. Dynamic Programming

A given sentence consists of n syllables $\mathrm{C = c_1c_2 \dots c_n}$

Normally, the longest Vietnamese word contains four syllables. The dynamic programming method is described in the following steps:

Step 1: Separate sentence C into combinations of one, two, three and four syllables.

Step 2: Calculate MI value for each combination of syllables.

Step 3: Calculate MI-score of final solution by following sub-steps:

1) Assume x is a chunk of syllables. f(x)=-100 if x is not in dictionary; whereas f(x)=MI(x). We select a value of -100 or lower in order to eliminate the solution that contains word outside dictionary.

2) Then calculate highest MI-score D[n] of final solution as following:

$\mathrm{D[0] = 0}$

$\mathrm{D[1] = f(c_1)}$

$\mathrm{D[2] = max\{ D[1] + f(c_2), D[0] + f(c_1c_2)\}}$

$\mathrm{D[3] = max\{ D[2] + f(c_3), D[1] + f(c_2c_3), D[0 + f(c_1c_2c_3)\}}$

$\mathrm{D[j] = max\ \{D[j\text{-}1] + f(c_j), D[j\text{-}2] + f(c_{j-1}c_j), D[j\text{-}3] + f(c_{j-2}c_{j-1}c_j), D[j\text{-}4] + f(c_{j-3}c_{j-2}c_{j-1}c_j)\}}$

With j = 4, 5, ..., n.

Step 4: After computing MI-score, the final segmentation is found by following sub-steps.

1) Set `K[j] = t`

and $\mathrm{D[j] = max\ \{D[j\text{-}1] + f(c_j), D[j\text{-}2] + f(c_{j-1}c_j), D[j\text{-}3] + f(c_{j-2}c_{j-1}c_j), D[j\text{-}4] + f(c_{j-3}c_{j-2}c_{j-1}c_j)\} = D[j\text{-}t] + f(c_{j-t+1} \dots c_j)}$

Where j-t is an index which maximizes the MI-score; and t value shows the best separated points in chunk of $(c_1 c_2 \dots c_j)$

$$c_1 c_2 \dots c_{j-t} \mid c_{j-t+1} \dots c_j \sim c_1 c_2 \dots c_{j-K[j]} \mid c_{j-K[j]+1} \dots c_j$$

The segmentation solution is $c_1 c_2 \dots c_{n-K[n]} \mid c_{n-K[n]+1} \dots c_n$

2) Set j = n-K[n]. The next segmentation is $c_1 c_2 \dots c_{j-K[j]} \mid c_{j-K[j]+1} \dots c_j$

Step 5: Loop step 4 until reach the first syllable in the sentence.

The actual calculation using dynamic programming method is following:

```
for (int i = 4; i <= n; i++)
{
   t1 = tudon[i - 2] + " " + tudon[i - 1];
   t2 = tudon[i - 3] + " " + tudon[i - 2] + " " + tudon[i-1];
   t3 = tudon[i - 4] + " " + tudon[i - 3] + " " + tudon[i - 2] +
" " + tudon[i-1];
   D[i] = max4(D[i - 1] + f[tudon[i-1]], D[i - 2] + f[t1], D[i -
3] + f[t2], D[i - 4] + f[t3]);
   maxMI = D[i];
   if (maxMI == (D[i - 1] + f[tudon[i - 1]]))
    {
      K[i] = i - 1;
    }
    else
          {
             if (maxMI == (D[i - 2] + f[t1]))
             {
                K[i] = i - 2;
             }
             else
             {
                if (maxMI == (D[i - 3] + f[t2]))
                {
                   K[i] = i - 3;
                }
                else
                {
                   K[i] = i - 4;
                }
             }
          }
       }
```

This method will be demonstrated by an example shown below:

Given a sentence in Vietnamese: *tôi lao động chăm chỉ*. (I word hard). The combinations of syllable: one-syllable- (tôi), (đi), (học), (chăm), (chỉ); two-syllables - (tôi lao), (lao động), (động chăm), (chăm chỉ); three-syllables - (tôi lao động), (lao động chăm), (động chăm chỉ); four-syllables - (tôi lao động chăm), (lao động chăm chỉ).

Using the proposed method, we can compute the highest MI-score and then get the separated points to segment the sentence. Figure 1 shows the programming results:

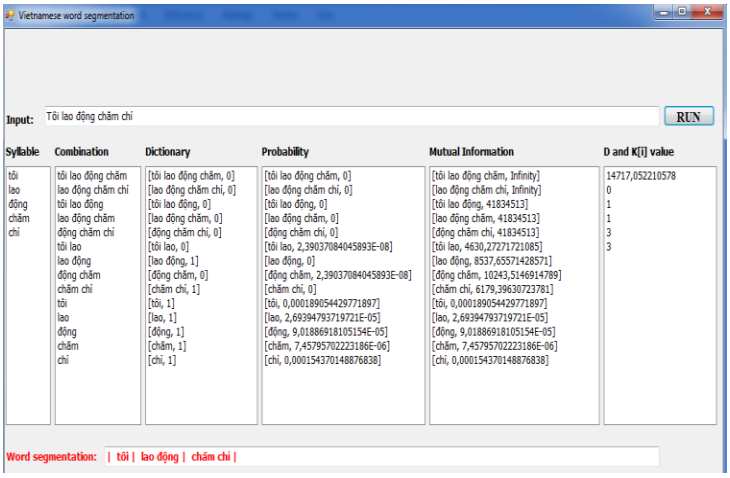


Fig. 1. Experiment with Vietnamese Sentence.

IV. Experimental Results

We extracted randomly 100 sentences from the corpus and asked 20 native Vietnamese speakers to make the word segmentation manually. For each sentence, we choose a solution which is selected by the most native speakers. The evaluation process is then taken by computing the rate as follows:

$$R = A/B$$

where:

A. Number of correct words which segmented by proposed method

B. Total number of words segmented by native speakers.

The rate is about 90%. This result is not very disappointing because we used only the raw corpus without lexicon nor annotation. We did not extend our experiment because we realized that we were not able to get better results with this method alone. However, the results show that the method works.

V. Conclusion

The proposed method has produced a promising results in the case of using un-annotated corpus for word segmentation. The mutual information is a key value to select the final segmented solution. The dynamic programming method is proposed to reduce the complexity of the problem. The advantage of our proposed method is that we do not need an annotation corpus. Therefore, the arbitrary text on the internet can be used as a corpus for natural language processing.

References

[1] Pak-kwong Wong, Chorkin Chan,"Chinese word segmentation based on maximum matching and word binding force". COLING '96 Proceedings of the 16th conference on Computational linguistics - Volume 1Pages 200-203, 1996.

[2] Dinh Dien, Hoang Kiem, Nguyen Van Toan., "Vietnamese Word Segmentation", The sixth Natural Language Processing Pacific Rim Symposium, Tokyo, Japan, pp. 749 -756, 2001.

[3] Pham DD., Tran GB., Pham SB., "A hybrid approach to Vietnamese word segmentation using part of speech tags", International Conference on Knowledge, 2009.

[4] Nguyen, P.T., Nguyen, V.V., Le, A.C., "Vietnamese word segmentation using hidden markov model", International Workshop for Computer, Information, and Communication Technologies on State of the Art and Future Trends of Information techonologies in Korea and Vietnam, 2003.

[5] Luu, T.A, Yamamoto, K., "A pointwise approach for Vietnamese Diacritics Restoration", IALP 2012.

[6] Le, H.P, Nguyen, T.M.H, Azim Roussanaly, Ho, T.V, "A hybrid approach to Word Segmentaion of Vietnamese texts", Language and automata theory and applications 2nd international coference, LATA 2008