Optimized Classification Rule Set Generation Using Pareto Based Binary Coded Elitist Multi Objective Genetic Algorithm

Dipankar Dutta* and Paramartha Dutta**
* Department of CSE, U.I.T., The University of Burdwan, Burdwan-713104, West Bengal, India.
** Department of CSS, Visva-Bharati University, Santinikatan-731235, West Bengal, India.

Abstract— **Data Mining is the process of extracting previously unknown, valid and actionable information from large databases and then using the information to make crucial decisions. One of the most important tasks in data mining is mining rules from set of data. Classification rule mining is one type of rule extraction process where rules can be represented in IF<Antecedent>THEN<Consequent> form. Where consequent part represent class label when conditions in the antecedent part is true. Classification rule mining problem is a multi objective problem (MOP) rather than a single objective problem (SOP). In this article we are using Multi Objective Genetic Algorithm (MOGA) to solve MOP. We are using two measures confidence and coverage for evaluating a rule. These measures can be thought of as different objectives of MOGA. Using these measures as the objectives of rule mining, we are extracting optimized rule set from different set of data using pareto based elitist MOGA abbreviated as CRGMOGA (Classification Rule Generation by Multi Objective Genetic Algorithm).**

I. INTRODUCTION

Genetic Algorithms (GAs) are "search algorithms based on the dynamics of natural selection and natural genetics" [1]. GAs can be broadly divided into two categories - Single Objective Genetic Algorithm (SOGA) and Multi Objective Genetic Algorithm (MOGA). When an optimization problem having objective function, the task of finding the best solution considering one objective is called Single Objective Optimization Problem (SOOP). But in the real world, life appears to be more complex. In fact most of the problems are having more than one objective, which are to be minimized or maximized simultaneously. These types of problems are called Multi-Objective Optimization Problem (MOOP) [2]. Classification rule mining problem is a MOOP. This works present a system based on MOGA to find out classification rule set. The use of GAs in classification is an attempt to exploit effectively the large search space usually associated with classification tasks [3] and better interactions among attributes. The knowledge discovery paradigms mostly used in data mining is yet rule induction. Most of the algorithms in this paradigm resort to local search [3]. In classification problem records are assigned to one among a small set of pre-defined classes.

This paper is organized as follows: Section II describes some of the steps for data preprocessing. Section III describes approaches of encoding and decoding the chromosome. Section IV defines MOOP and pareto approaches of MOGA for solving MOOP. Section V describes how we are applying MOGA in classification rule mining problem. Section VI illustrates algorithms of processes. Section VII describes how we are measuring objectives in the context of whole dataset and comparing our method with other existing methods. Finally section VIII ends the paper along with future directions.

II. DATA PREPROCESSING

To make quality decision, quality data are required. Data preprocessing is done which includes the following steps [5], [6], [7] to improve the quality of the data such as accuracy, completeness and consistency etc.

A. Data Integration

This is necessary if data such as multiple databases, data cubes, or files to be mined come from several different sources, such as several departments of an organization. This step involves, for instance, removing inconsistencies in attribute names between data sets of different sources. As we are using data from UCI machine learning repository [8], this step is not relevant to this work.

B. Data Cleaning

It is important to make sure that the data to be mined is as accurate as possible. This step may involve detecting and correcting errors in the data, filling in missing values, identifying or removing noisy data, outliers etc. In some of the datasets some records are having missing values in UCI machine learning repository. In this paper we have not reported any rule set found from datasets having missing values. If we consider datasets, which have some missing values, those are treated as one type of attribute value. We are not getting good results by this and in our future work we shall explore different methods to deal with missing values in dataset.

C. Conversion

For a categorical attribute, number of distinct values of that attribute form different categories of that attribute. Then categorical attributes are converted into continuous attributes by assigning different numeric values to different category. Minimum value is one and maximum value is equal to the

number of distinct values of that attribute. Now all attributes become continuous and can be treated in the same manner.

D. Discretization

This step reduces the number of values for a given continuous attribute by dividing the range of the attribute into intervals. It transforms a continuous attribute into a discrete attribute, taking only a few discrete values. An equal-width (distance) partitioning is used to divide the range into N intervals of equal sized: uniform grid. If MinValue and MaxValue are the lowest and highest values of the attribute, the width of intervals will be: rangeOfDivision = (MaxValue-MinValue)/N. Here N=2,4,8,16 or 32 based on value of (MaxValue-MinValue+1) i.e., range so that if we convert data into binary 1,2,3,4 or 5 bits respectively will be sufficient to store those data. In this way, continuous attribute can be presented in binary notation. How many bits will be used will depend on the range of values of the attributes. For range 2 we are using 1 bit, for range 3 and 4 we are using 2 bits, for range 5 to 8 we are using 3 bits, for range 9 to 16 we are using 4 bits, for others we are using 5 bits.

E. Attribute Selection

This step consists of selecting a subset of attributes relevant for classification among all original attributes.

Attribute selection methods can be divided into filter and wrapper approaches. In this work we can use filter approach in which we are considering those attributes only that are relevant in the context of classification. Other attributes are eliminated.

For example in Zoo dataset from UCI machine learning repository we are eliminating animal name that is unique for each record and will not help in any way to extract rules for classification before running MOGA.

If we keep attributes like this (animal name in Zoo dataset) MOGA will not generate rule with these attributes. Measures used for selecting rules will eliminate rules having these types of attributes following wrapper approach. Here computational load is higher than that in filter approach. So whenever possible we are following filter approach.

III. CHROMOSOME ENCODING AND DECODING

GAs for rule discovery can be divided into two broad approaches, based on how rules are encoded in the population of individuals ("chromosomes"). They are Pittsburgh approach and Michigan approach. In our work we have followed Michigan approach, where each individual encodes a single classification rule representing one class label. We are measuring accuracy of the rule set by applying all rules on each record.

Data may be of two types - categorical and numerical. As discussed in section II.C, categorical data can be converted into numerical data and can be dealt subsequently.

A chromosome may have n genes, where each gene corresponds to one attribute, and n is the number of antecedent attributes in the data being mined. A numeric attribute is represented by a set of Binary-coded lower and upper limits, where each limit is allocated an user-defined number of bits (noOfBits) [9]. How we are deciding about the noOfBits is discussed in section II.D. Here number of division

$$\text{numberOfDivision}=2^{\text{noOfBits}} \quad (1)$$

Range of values in a division

$$\text{rangeOfDivision}=(\text{MaxValue}-\text{MinValue})/\text{numberOfDivision} \quad (2)$$

A numeric value is converted into binary by converting (Value-MinValue)/ rangeOfDivision into binary. If numbers of bits produced are lower than noOfBits we are padding more significant bits with 0.

When the string representing a rule is decoded, we are following the reverse procedure by using MinValue, noOfBits and rangeOfDivision. The procedure is

$$\text{Value}= \text{MinValue} + (\text{rangeOfDivision})*\left(\sum_{i=1}^{i=\text{noOfBits}} (2^{i-1}*i^{th} \text{ bit value})\right) \quad (3)$$

All genes are positional, i.e., the first gene represents the first attribute, the second gene represents the second attribute had chosen for building chromosome, and so on, as we are selecting attributes for a particular generation of GA. By this we can avoid generation of invalid rule after crossover (Described in section V.C). Each gene corresponds to one attribute in the IF part of a rule, and the entire chromosome (individual) corresponds with the entire IF part of the rule. The THEN part does not need to be encoded into the chromosome.

For each gene representing one attribute, has two parts. First part represents minimum value of the attribute and second part represents maximum value of the attribute.

$$\text{Length of the chromosome} = 2*\left(\sum_{i=1}^{i=\text{ no of Attributes}} (\text{Bit length of attribute})\right) \quad (4)$$

Chromosome lengths are same for all chromosomes for a particular run of GA but we may choose different combination of attributes for different run of GA. So length of chromosome is varying over runs. Hence, different individuals correspond to rules with different number of predicting attributes. Since length of all chromosomes in a particular run of GA is same, we can avoid complexity of designing GA operator such as crossover operator for handing variable length chromosome.

IV. MOOP AND PARETO APPROACH

We state the MOOP in its general form [2]:

Maximize/Minimize $f_m(x)$, $m=1,2,...,M;$

Subject to: $g_j(x)\geq 0$, $j=1,2,...,J;$

$h_k(x)=0$, $k=1,2,...,K;$

$x_i^{(L)} \leq x_i \leq x_i^{(U)}$, $i = 1,2,...,n;$

There are three different approaches to cope with MOOP by MOGA i) weighted sum approach ii) the lexicographical approach, and iii) the pareto approach.

In this work we are following pareto approach. The basic idea of pareto approach is that, instead of transforming a MOP into a SOP and then solving it by GA, one should use MOGA directly. Adopt the algorithm to the problem being solved, rather than the other way around. In this context we have to define pareto dominance. As we are extracting rules we should

define it in terms of rule. A rule r_1 is said to dominate another rule r_2 iff r_1 is strictly better than r_2 with respect to at least one criterion (objectives or measures) being optimized and r_1 is not worse than r_2 with respect to all the criteria being optimized. Using the pareto dominance, rules are compared among one another, i.e., a rule is dominant over another only if it has better performance in at least one criterion and non-inferior performance in all criteria. A rule is said to be pareto optimal if it cannot be dominated by any other rule in the search space. In complex search spaces, wherein exhaustive search is infeasible, it is very difficult to guarantee pareto optimality. Therefore instead of the true set of optimal rules (pareto set), one usually aims to derive a set of non-dominated rules with objective values as close as possible to the objective values (pareto front) of the pareto set [10].

V. CLASSIFICATION RULE MINING USING MOGA

A classification rule is represented in the IF<antecedent> THEN<consequence> structure. Here antecedent represents some conditions to be satisfied and consequence predicts class label. In this article, antecedent and consequence will be referred as A and C respectively unless otherwise mentioned.

A. Objectives of MOGA

We are considering two measures for evaluating fitness of a classification rule. They are confidence and coverage. Maximizing them simultaneously are the objectives of MOGA.

1) Confidence

This objective gives the accuracy of the rules extracted from the dataset.

$$\text{Confidence}=\text{SUP}(A\&C)/\text{SUP}(A) \quad (5)$$

SUP(A&C) is the number of records that satisfy both antecedent A and consequent C.

Whereas SUP(A) is the numbers of records satisfy all conditions in antecedent A.

2) Coverage

This measure is defined here as the proportion of the target class covered by the rule

$$\text{Coverage}=\text{SUP}(A\&C)/\text{SUP}(C) \quad (6)$$

SUP(C) is the number of records having class label as in consequent C.

Classification rule mining problem becomes a two objective optimization problem where we have to maximize confidence and coverage. As we are following elitist MOGA [2] we are selecting chromosomes (rules) based on their values of confidence and coverage.

B. Assigning Class Labels

Chromosomes are encoding antecedent part of the rules. Consequent part is not required to be coded into chromosome. Predicted class can be the chosen that has more representative in the set of records satisfying the rule antecedent [11]. For classes, which have lower instances in the dataset, we will not get enough support counts for those classes and as a result we shall not get rules for those classes. This is a major problem.

The above-mentioned problem can be solved by the following way. For a particular set of antecedent attributes in a rule, we are calculating support count for every class. Then we are dividing support count for each class with the number of instance of that class in the dataset. We are assigning class label to that rule, which class is giving highest value of (support count/number of instance of class) (Explained in section VI.A).

C. Crossover and Mutation

In all our experiments probabilities of crossover and mutation are fixed at 0.5 and 0.1 respectively. Crossover and mutation operation may be single point or multi point. In case of crossover we are selecting two parent chromosomes randomly from population. We are selecting a position randomly to do crossover for producing child chromosomes. Child chromosomes are replacing parent chromosomes in the population.

D. Population Size and Selection Procedure

Before first run and for every alternative run of GA 10% records from dataset are randomly selected for creation of populations. Random combination of attributes is chosen for building antecedent part of the rules. Minimum and maximum values of all attributes of a rule is same as the attribute values of the record. This generates rules with different combination of attributes and rules lying in the pareto front survive. We have also used an elitist reproduction tactics, where rules lies on the pareto optimal front among child and parent population of every class after each generation was passed unaltered to the next generation. So population size is allowed to vary.

For subsequent alternative run, uniform random selection was used to select a chromosome from the available set produced after elitist reproduction. This will reduces the selection pressure, since roulette wheel selection resulted in premature convergence (at the start of the run all rules performed very poorly as few rules matched any instance) [4]. Other chromosomes are selected which are having valid values for same attributes as of the first chromosome selected. So for a particular generation of GA, length of the chromosome is fixed but for different generation of GA, length of the chromosome is varying, as number of attributes selected as antecedent is varying. When chromosome with valid values for all attributes are selected, random combination of attributes are used to build chromosomes for next generation of GA.

E. Stopping Condition

In this work the stopping condition was carried out as a prefixed number of generations set by the user.

VI. ALGORITHMS

A. Assigning Class Labels

1) For each rule, count number of instances, where Antecedent or A is true for every class and denote it as A_{ij} where i=index of rules and j=index of classes.

2) Count number of instances of every class and denote it as C_j where j=index of classes.

3) For each rule, calculate A_{ij}/C_j =(Number of instances where A is true of i^{th} rule and j^{th} class)/(Number of instances of j^{th} classes).

4) Assign class label as C_i to the rule for which class A_{ij}/C_j is highest where i=index of rules.

B. Calculating Objectives

1) For each rule, count the number of instances, where Antecedent or A is true. Denote it as A_i where i=index of rules.

2) For each rule, count the number of instances of labeled class and denote it as C_i where i=index of rules.

3) For each rule, count the number of instances, where A and C are true and denote it as $A\&C_i$ where i=index of rules.

4) For each rule, calculate confidence $A\&C_i/A_i$ and denote it as $Confidence_i$ where i=index of rules.

5) For each rule, calculate coverage $A\&C_i/C_i$ and denote it as $Coverage_i$ where i=index of rules.

C. Combination

1) Select all rules lie on the pareto optimal front after previous generation of GA.

2) Select all rules lie on the pareto optimal front after Selection,Crossover, Mutation, Step A and Step B.

3) Combine selected rules.

D. Elite Selection

1) Find out number of classes.

2) Initialize j=1.

3) If j≤(number of classes) then for j^{th} class select distinct rules from Combined rules and go to step 5.

4) Else go to step 21.

5) Find out total number of selected rules and denote it as numberOfRules.

6) If numberOfRules=1 for any class then select that rule.

7) Increment j by one and go to step 3.

8) Else if numberOfRules>1 then initialize i=1.

9) If i≤ numberOfRules select i^{th} rule and go to step 11.

10) Else go to step 20.

11) Initialize k=1.

12) If k≤numberOfRules select k^{th} rule then go to step 14.

13) Else go to step 18.

14) If k≠i then go to step 16.

15) Else increment k by one and go to step12.

16) If($(Confidence_k>Confidence_i AND Coverage_k>Coverage_i)$ $OR(Confidence_k==Confidence_i AND Coverage_k>Coverage_i)$ $OR \quad (Confidence_k>Confidence_i AND Coverage_k==Coverage_i)$) set flag=false, increment i by one and go to step 9.

17) Else set flag=true, increment k by one and go to step 12.

18) If flag=true select i^{th} rule and increment i by one then go to step 9.

19) Else If flag=false increment i by one and go to step 9.

20) Increment j by 1 and go to step 3.

21) Stop.

E. Selection

1) Before the start of next run of GA, select all rules from previous generation of GA.

2) Select any rule from parent rules.

3) Initialize i=1.

4) If i≤ (number of attributes) go to step 6.

5) Else go to step 9.

6) If value of i^{th} attribute is not valid columeVariables[i]=NO.

7) Else value of i^{th} attribute is valid columeVariables[i]=YES.

8) Increment i by one and go to step 4.

9) Initialize j=1.

10) If j≤ (number of rules in Servived after previous generation of GA) go to step 12.

11) Else go to step 21.

12) Set i=1;

13) Set Flag=true;

14) If i≤ (number of attributes) go to step 19.

15) Else for j^{th} rule If value of i^{th} attribute is not valid and columeVariables[i]=YES set Flag=false.

16) For j^{th} rule If value of i^{th} attribute is valid and columeVariables[i]=NO set Flag=false.

17) If Flag=false increment j by 1 and go to step 10.

18) Else If Flag=true increment i by 1 and go to step 14.

19) If Flag=true then select j^{th} rule from rules of previous generation of GA as parent rule of this generation of GA, increment j by 1 and go to step 10.

20) Else If Flag=false increment j by 1 and go to step 10.

21) Count the number of parent rules.

22) If number of rules is less than 2 select 10% records as rules from original data.

23) Else stop.

VII. MEASURES AND COMPARISON

We have carried out experiments on eight-dataset form UCI Repository of Machine Learning Databases [8] as listed in TABLE1. These dataset do not have any missing value, which is one of the fundamental reasons for choosing them.

A. Measuring Objectives

For each rule, we are trying to maximize two measures confidence and coverage simultaneously. Each record in the dataset is checked with all rules generated. For a particular record and a particular rule, for each attribute of the record, we are checking whether the value of the attribute is within the range specified by maximum and minimum value of that attribute in the antecedent part of the rule. If for all attributes in the antecedent part of the rule, a record's corresponding attribute value fall within the range then that rule can be used to classify that record. By applying all generated rules on a record if a single rule found is satisfying the above-mentioned criteria, we may uses that rule for classifying that record. If more than one rules are found satisfying the above-mentioned criteria, we are choosing rule, with maximum confidence value. If more than one rules are having same confidence we are picking up rule with highest coverage. If more than one rules are giving

same confidence and coverage we are selecting a rule randomly. In that manner we are classifying all record by using a particular rule for classification. If there is a match between actual class label and predicted class label the record is correctly classified, otherwise it is misclassified. So in the context of the whole dataset we can measure accuracy of the rule set produced by MOGA by the following

Accuracy in %=(Number of matches*100)/Number of records (7)

Now it may also happen that for a particular record no rule is covering that record. In that case the second objective is not fulfilled. So in respect of the whole dataset, we can measure coverage of the rule set by the following

Coverage in %= 1-(Number of records when rule not Found*100/Number of records) (8)

B. *Comparison with Other Methods*

TABLE I. THE COMPARISON OF C4.5, CBA, CMAR AND CRGMOGA ON CONFIDENCE

Dataset	C4.5 (%)	CBA(%)	CMAR(%)	CRGMOGA(%)
Glass	68.7	73.9	70.1	71.5
Ionosphere	90	92.3	91.5	89.1
Iris	95.3	94.7	94	95.3
Pima	75.5	72.9	75.1	74.9
Tic Tac Toe	99.4	99.6	99.2	73.6
Waveform	78.1	80	83.2	75.2
Wine	92.7	95	95	93.8
Zoo	92.2	96.8	97.1	100

In this section, we are comparing our method CRGMOGA with three popular classification methods: C4.5 [12], CBA [13] and CMAR [14]. CRGMOGA's performance is not up to the mark in case of Tic Tac Toe dataset only as described in Table 1. For other cases it is performing better or its performance is comparable with other methods.

Another objective of CRGMOGA is maximizing coverage. Rules generated by CRGMOGA in cases of all 8 datasets gives 100% coverage. Literatures [12], [13], [14] describing other methods are not giving any measures like this.

VIII. CONCLUSIONS AND FUTURE DIRECTIONS

This paper illustrates classification rule mining method by using MOGA. A set of rules with high confidence and coverage value is generated. In future algorithms can be improved to prune unnecessary rules. More elaborated experiments to optimize several parameters like crossover and mutation rate is necessary. Parallelization of this algorithm can be done. This method of rule finding can deal with datasets having missing values by considering them as one type of attribute value. Well-known methods of substitution of missing values can be applied to improve performance of CRGMOGA.

REFERENCES

[1] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st ed., Boston, MA, USA, Addison-Wesley Longman, 1989.

[2] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Chichester, UK, John Wiley and Sons, 2001.

[3] M. V. Fedelis, H. S. Lopes, and A. A. Freitas, "Discovering Comprehensible Classification Rules with a Genetic Algorithm," Proc. of the Cong. on Evolutionary Computation, San Diego, USA, pp. 805-810, 2000.

[4] A. L. Corcoran and S. Sen, "Using Real-Valued Genetic Algorithms to Evolve Rule Sets for Classification," Proc. of the Cong. on Evolutionary Computation, Orlando, Florida, USA, pp. 120-124, 1994.

[5] J. Han, and M. Kamber, Data Mining: Concepts and Techniques, 2nd ed., J. Gray, Eds. San Francisco, CA, USA, Morgan Kaufmann, 2006.

[6] D. Pyle, Data Preparation for Data Mining, San Francisco, CA, USA, Morgan Kaufmann, 1999.

[7] A. A. Freitas, A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery, Postgraduate Program in Computer Science, Pontilicia Universidade Catolica do Parana Rna Imaculada Conceicao, 1155, Curitiba - PR, 80215-901, Brazil , 2000.

[8] UCI Repository of Machine Learning Databases, Department of Information and Computer Science, University of California, Irvine, CA, USA,[Online],Available:http://www.ics.uci.edu/~mlearn/MLRepository.html.

[9] B. de la. Iglesia, M. S. Philpott, A. J. Bagnall, and V.J. Rayward-Smith, "Data Mining Rules Using Multi-Objective Evolutionary Algorithms," Proc. of the Cong. on Evolutionary Computation, Canberra, Australia, vol. 13, pp. 1552-1559, 2003.

[10] A. A. Freitas, "A Critical Review of Multi-Objective Optimization in Data Mining: a Position Paper," ACM SIGKDD Explorations Newsletter, vol. 6, no. 2, pp. 77-86, 2004.

[11] A. Giordana, and F. Neri, "Search-Intensive Concept Induction," Evolutionary Computation, vol. 3, no. 4, pp.375-416, 1995.

[12] J. R. Quinlan, C4.5: Programs for Machine Learning, San Mateo, CA, USA, Morgan Kaufmann, 1993.

[13] B. Liu, W. Hsu, and Y. Ma, "Integrating Classification and Association Rule Mining," Proc. of the 4th Int. Conf. on Knowledge Discovery and Data Mining , New York, USA, pp.80-86, 1998.

[14] W. Li, J. Han, and J. Pei, "CMAR: Accurate and Efficient Classification based on Multiple Class-Association Rules." Proc. of the 1st IEEE Int. Conf. on Data Mining ,San Jose, CA, USA, pp. 369-376, 2001.

Dipankar Dutta received his Bachelor of Engineering degree in the year of 1996 from Jalpaiguri Government Engineering College, The University of North Bengal, India. In 2000 he completed PGDIM from IGNOU, India. In 2004 he received his M.Tech in Computer Technology from Jadavpur University, India.

During 1996 to 2003 he worked at various industrial houses like Brite Metalloy Pvt. Ltd. and D. C. Industrial Plant Services Pvt. Ltd. From 2004 he worked in Haldia Institute of Technology, Asansol Engineering College, University Institute of Technology. Presently he is serving as Assistant Professor in Computer Science and Engineering Department of University Institute of Technology, The University of Burdwan, West Bengal, India. His fields of interest are Data Mining, Soft Computing etc.

Dr. Paramartha Dutta did his Bachelors and Masters in Statistics from the Indian Statistical Institute, Calcutta in the years 1988 and 1990 respectively. He afterwards completed his Master of Technology in Computer science from the same Institute in the year 1993 and Doctor of Philosophy in Engineering from the Bengal Engineering and Science University, Shibpur in 2005 respectively.

He has served in the capacity of research personnel in various projects funded by Govt. of India, which include Defence Research and Development Organization, Council of Scientific and Industrial Research, Indian Statistical Institute, Calcutta etc.

Dr. Dutta is now a Professor in the Department of Computer and system Sciences of the Visva Bharati University, West Bengal, India. Prior to this, he has served Kalyani Government Engineering College and College of Engineering and Management, Kolaghat, both in West Bengal as full time faculty members. Apart from this, he has remained associated to National Institute of Management, Calcutta, Department of Statistics of University of Calcutta, Department of Computer Science and Technology of Bengal Engineering and Science University, Shibpur either as a guest faculty member from time to time.

Dr. Dutta has been involved as the principal investigator funded by the All India Council for Technical Education, Govt. of India. He has co-authored four books and has also one edited book to his credit. He has published about ninety papers in various journals and conference proceedings, both international and national. Presently, he is supervising three students for their Ph. D programme registered with either Visva Bharati University or West Bengal University of Technology.

Dr. Dutta is a Life Fellow of the Optical Society of India (OSI), Life Member of Institution of Electronics and Telecommunication Engineers (OSI), Computer Society of India (CSI), Indian Science Congress Association (ISCA), Indian Society for Technical Education (ISTE), Indian Unit of Pattern Recognition and Artificial Intelligence (IUPRAI) - the Indian associate of the International Association for Pattern Recognition (IAPR), Member of Associated Computing Machinery (ACM).