Proceedings of International Conference FACT-2009:
Recent Trends in Computing and Communications, December 18-19, 2009:
K C G College of Technology, Chennai, TN, INDIA:

Discovering Prediction Rules for Predicting Quality of Students Using Multi Objective Genetic Algorithm from Student Database

Dipankar Dutta*
Department of Computer Science and Engineering,
University Institute of Technology, The University of Burdwan, Golapbug (North), Burdwan, West Bengal, PIN-713104, India.
Email: dipankar_dutta@rediffmail.com

***Abstract*- Machine Learning tries to induct knowledge automatically from a set of records or instances of a problem. Discovering Prediction Rules is one of the knowledge discovering process in Machine Learning where rules of the form IF-THEN structure. IF<Some condition is satisfied>THEN<Predict class label>. IF part is termed Antecedent (A) and those with the THEN part is called Consequence (C).**

There are many different approaches of learning prediction rules from examples. Many of them strive to obtain the best possible accuracy without bothering about the comprehensibility of the knowledge gained, but a set of Prediction Rules must satisfy some properties. Rules should be accurate as far as possible, they should cover records in the databases as far as possible and they should be as simple as possible. Maximizing these three objectives of rule mining problem can be extracted by using Multi-Objective Genetic Algorithms (MOGAs).

***Index Terms*- Binary coded, lexicographical approach, multi objective genetic algorithm, prediction rules.**

I. INTRODUCTION

Data mining emerged from the need for converting records stored in large databases into useful, interesting and comprehensible knowledge. Data mining is typically performed on real-world databases that had been created for purposes other than learning [10].

In this work MOGA is used to find out prediction rules. In essence, Genetic Algorithms (GAs) are "search algorithms based on the mechanics of natural selection and natural genetics" [8]. One advantage of GA over "traditional" search methods is that the former performs somewhat a global search using a population of individuals, rather than performing a local, hill-climbing search. Global search methods are less likely to get trapped into local maxima, in comparison with local search methods. It is interesting to note that, overall, the knowledge discovery paradigm most used in data mining is still rule induction. Most of the algorithms in this paradigm perform somewhat a local search [4].

II. DESCRIPTION OF DATA SET USED

We are using records of students of our college. Student database having 21 attributes naming Roll Number, Department, Name, Birth Year, Contact No, Tenth Percentage, Year of Passing Tenth, Twelfth Percentage, Year of Passing Twelfth, Medium of study, First Semester Percentage, Second Semester Percentage, Third Semester Percentage, Fourth Semester Percentage, Fifth Semester Percentage, Sixth Semester Percentage, Seventh Semester Percentage, Eighth Semester Percentage, Graduation Average Percentage, Number of

Campus interviews attended, Company names. Out of these attributes Roll Number, Department, Name, Contact No, Medium of study and Company names are categorical and remaining attributes are continuous.

III. DATA PREPROCESSING STEPS

It includes the following steps (among others) [13]:

1) Data Integration: This is necessary if the data to be mined comes from several different sources, such as several departments of an organization. Here we have collected data about students from administrative, examination and placement department of our college.

2) 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, etc. Data cleaning has a strong overlap with data integration. Here we have deleted student's record who did not complete their study and having unknown values in there record. After this step database having records of 161 students.

3) Classification: Here we are using Number of Campus interviews attended and Company names for classifying students at different class. We are assigning 8 (0-7) different class labels to different students based on the number of campus interview attended by them before getting job and the number of campus interview attended by them but failed to got job.

4) Attribute Selection: Between filter and wrapper approaches [12] following filter approach student's name, roll no and contact no are eliminated from database as they do not have any power of classifying students. This will reduces computational load. Among other attributes Number of Campus interviews attended and Company names are not used for predicting class label because we are using them for classifying students into different class.

5) Categorical to Continuous: Among 16 attributes are used in the antecedent part; Department and Medium of study are categorical. We are converting categorical attributes into continuous by assigning unique integer values to each categories starting from integer 0. Then all attributes can be treated as continuous attribute.

6) Discretization: This step consists of transforming a continuous attribute into a binary, taking only a few discrete values. We have calculated maximum and minimum values of an attribute to find the range of that attribute. Then we are dividing the range into 2, 4 or 8 divisions so that if we convert data into binary and 1, 2 or 3 bits respectively will be suitable to store that datum. We are restricting no of divisions to 8 for restricting no of bits used per attribute to 3. We can go for higher no of divisions and higher no of bits but in that case very few numbers of instances will fall in each division and length of the chromosome will also increase. As we are working with continuous attributes having integer values only, many attributes having range 2 or 4 where we can use 1 or 2 bits respectively. After conversion from integer to binary if number of bits required for representing a value is less than the maximum no of bits required for representing that attribute, we are padding higher order bits by 0. This will help to keep length of the chromosomes same when we are constructing them from a set of attributes.

IV. RULES FORMATION AND SELECTION

A. Individual Representation

GA for rule discovery can be divided into two broad approaches, based on how rules are encoded in the population of individuals ("chromosomes"). In the Michigan approach [9] each individual encodes a single prediction rule, whereas in the Pittsburgh approach [14] each individual encodes a set of prediction rules. In our work we have followed Michigan approach where each rule predicts one class of students.

B. Individual Encoding

A chromosome is divided into n genes, where each gene corresponds one attribute and n is the number of predicting or antecedent attributes in the data being mined.

The genes are positional, i.e., the first gene represents the first attribute, and the second gene represents the second attribute had chosen at a particular run of GA and so on. By this we can avoid generation of invalid data after crossover. Each gene corresponds to one attribute in the IF part of a rule, and the entire chromosome (individual) corresponds to the entire IF part of the rule. The THEN part does not need to be coded into the chromosome.

Here maximum value of n is 16. At a particular run of GA all 16 attributes may be chosen or any combinations these attributes may be chosen. Length of the chromosome can be obtained by adding bit lengths of individual genes. Lengths of chromosomes are same for a particular run of GA but we may choose different combinations of attributes for different runs of GA. So length of chromosomes are varying from generation to generations of GA but length of all chromosomes are same for a particular generation of GA. By this we can avoid complexity of designing GA operator such as crossover operator for handing variable length chromosomes.

B. Selection Process

Before first run of GA 20 initial records are selected randomly from database. Binary values of all 16 attributes are taken for constructing 20 chromosomes of GA. So initial population size is 20. In the subsequent runs of GA before each run of GA following procedure has been used for constructing population of chromosomes.

1) Select any chromosome from chromosomes survived after previous run of GA. 2) If all 16 attributes of selected chromosomes having valid values then select a random combination of attributes. 3) Construct chromosomes with that combination of attributes from chromosomes survived after previous run of GA having valid values for all 16 attributes. 4) If all 16 attributes of selected chromosomes do not have valid values construct two separate attributes set one having valid values and other for invalid values. 5) Select chromosomes having valid and invalid values for same set of attributes from chromosomes survived after previous run of GA. 6) If no of chromosomes selected<2 then select 20 initial records randomly from database for constructing chromosomes.

For crossover at least 2 chromosomes are required. Uniform random selection method of selecting the first chromosome will reduce 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) [1]. At the end of each generation parent chromosomes and child chromosomes are combined and some top ranked chromosomes are chosen for next generation of GA following elitist reproduction strategy [2]. From those rules we are selecting some of them for constructing population for the next generation. So population size is varying from generation to generation of GA.

V. GA OPERATIONS PARAMETERS AND STOPPING CONDITION

Here we are using single point crossover with crossover rate of 50% fixed and mutation rate of 10% fixed.

The user can set prefixed number of generations that can be used as the stopping condition of this algorithm. At the end of each generation we are measuring the number of rules produced. We can also stop when we are not getting new rules in consecutive large number of generations.

VI. MULTI-OBJECTIVE OPTIMIZATION AND RULE MINING

For a multi-objective problem in usually we will get a set of solution instead of a single solution by using MOGA. Multi-objective problem can be solved by MOGA by following three different approaches, 1) weighted sum approach where a multi-objective problem is transformed into a single objective problem by a weighted sum of objective functions 2) the Pareto approach which consists of as many non-dominated solutions as possible and returning the set of Pareto front to the user. A solution $s1$ is said to dominate (in the Pareto sense) a solution $s2$ if and only if $s1$ is strictly better than $s2$ with respect to at least one of the criteria (objectives) is being optimized and $s1$ is not worse than $s2$ with respect to all the

criteria had been optimized [6][15][2]. 3) the lexicographical approach, where the objectives are ranked in order of priority. The weighted sum approach-which is so far the most frequently used in the data mining literature-is to a large extent an ad-hoc approach for multi-objective optimization, whereas the lexicographical and the Pareto approaches are more principled approaches [6].

The basic idea of Lexicographical approach is to assign different priorities to different objectives and then focus on optimizing the objectives in their order of priority. The lexicographical approach has important advantage over the weighted sum approach: the former avoids the problem of mixing non-commensurable criteria in the same formula. Indeed, the lexicographical approach treats each of the criteria separately, recognizing that each criterion measures a different aspect of quality of a candidate solution. As a result, the lexicographic approach avoids the drawbacks associated with the weighted sum approach such as the problem of fixing weights.

Three objectives of MOGA can be defined as follow

$$\text{Confidence} = \frac{\text{SUP(AUC)}}{\text{SUP(A)}} \quad (1)$$

where SUP(AUC) is the support count when A and C is true and SUP(A) is the support count when A is true.

This measure represents the proportion of records for which the prediction of the rule is correct.

$$\text{Coverage} = \frac{\text{SUP(AUC)}}{\text{SUP(C)}} \quad (2)$$

where SUP(AUC) is the support count when A and C is true and SUP(C) is the support count when C is true.

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

$$\text{Attractiveness} = \frac{1}{\text{Number of attributes in A}} \quad (3)$$

The generated rule may have many attributes involved in the rule thereby making it difficult to understand [4]. Smaller rules are more attractive to the user than larger rules.

Prediction rule mining problem becomes a three objective optimization problem where we have to maximize confidence, coverage and attractiveness.

For a particular set of antecedent attributes in a rule we are calculating support count for each class and we are assigning class label to class that is giving highest support count [7]. Then we are calculating confidence, coverage and attractiveness of that rule. So for each generation for GA we are discovering rules predicting different classes unlike some of algorithm where each generation of GA are searching for rules predicting the same class [4].

Following lexicographical approach we are selecting rules by giving highest importance to confidence factor i.e., accuracy of the rules, then to coverage of the rules and lowest importance is given to attractiveness of the rules. Pseudo code for above-mentioned process is given below

```
for(Rank=0;Rank<15;Rank++)
{
for(ClassLabel=0;ClassLabel<=7;ClassLabel++)
{
MaxConfidence=-1;
MaxCoverage=-1;
MaxAttractiveness=-1;

SELECT Maximum Confidence AS
MaximumConfidence FROM
SelectionConfidenceCoverageSorted table
WHERE ClassOfStudent=ClassLabel;

if(No record for class ClassLabel of student)
{
Skip rest of the inner for loop;
}
else
```

```
{
MaxConfidence=MaximumConfidence;

SELECT Maximum Coverage AS
MaximumCoverage FROM
SelectionConfidenceCoverageSorted table
WHERE ClassOfStudent=j AND
Confidence=MaxConfidence;

MaxCoverage=MaximumCoverage;

SELECT Maximum Attractiveness AS
MaximumAttractiveness FROM
SelectionConfidenceCoverageSorted table
WHERE ClassOfStudent=j AND
Coverage=MaxCoverage AND
Confidence=MaxConfidence;

MaxAttractiveness=MaximumAttractiveness;

if(MaxConfidence!=0.0 AND
MaxCoverage!=0.0 AND
MaxAttractiveness!=0.0)
{
INSERT INTO EliteSelection table SELECT
(DISTINCT * FROM
SelectionConfidenceCoverageSorted table
WHERE Confidence=MaxConfidence AND
Coverage=MaxCoverage AND
Attractiveness=MaxAttractiveness AND
ClassOfStudent=ClassLabel);

DELETE FROM
SelectionConfidenceCoverageSorted table
WHERE Confidence=MaxConfidence AND
Coverage=MaxCoverage AND
Attractiveness=MaxAttractiveness
AND ClassOfStudent=ClassLabel;
}
}
}
}
```

VII. DISCOVERED-KNOWLEDGE POST PROCESSING

In our work we are comparing every rule with other rules of that class. If two rules varying in one attribute by value 1 we can merge two rules by assigning one minimum value and one maximum value for that attribute keeping other attributes value unaltered. This step is not required in rules generation process when a range presents every attribute i.e., minimum value and maximum value in the antecedent part of the rule during rules discovery process [3][1]. For categorical data we cannot specify minimum value or maximum value and therefore this type of rule formation process is not very suitable for categorical data. In this work as we are not storing minimum and maximum values for each attribute in rules during rules discovering process, this type of rules reduction process gives good result. This rules formation process keeps size of the rules smaller during generations of MOGA. In some of the rules mining process weights and relational operator also have to be encoded into chromosomes along with the values of attributes [4][5]. As we are encoding values of attributes only in chromosomes, chromosomes lengths becomes smaller.

VIII. RESULTS

For a record we are calculating distances for each rule and choosing rule that is giving lowest distance. If more than one rule are giving same distances we are choosing rule/or rules with higher confidence value for prediction. If more than one rule are having same confidence value, a rule is chosen randomly. In that manner we are predicting all records by using a rule for prediction. If actual class label and predicted class label is same then there is a match otherwise mismatch.

$$\text{Accuracy in } \% = \frac{\text{Number of matches} * 100}{\text{Number of records}} \quad (4)$$

$$\text{Coverage in } \% = \frac{\text{Number of records covered by rules} * 100}{\text{Number of records}} \quad (5)$$

$$\text{Average Attractiveness} = \frac{\text{Summation of Attractiveness of rules}}{\text{Number of rules}} \quad (6)$$

In one run after 2000 generations we are getting 98% accuracy, 100% coverage, 0.264 average attractiveness. We are measuring three objective factors of the rules generated after each 100 generations of GA and noticed that their values are increasing gradually. We are also getting new rules with the new generations of GA. For other runs we are getting results in the same range.

IX. CONCLUSION

This paper illustrates prediction rules mining method by using MOGA. A set of rules with high confidence, coverage and attractiveness values is generated. Rules are giving high accuracy rate while predicting class label. In future algorithms can be developed to prune rules without compromising for accuracy rate for predicting class label. More elaborated experiments to optimize several parameters like crossover, mutation rate and rank is necessary. Parallelizing of this algorithm can be done.

REFERENCES

[1] A. L. Corcoran, and S. Sen, "Using real-valued genetic algorithms to evolve rule sets for classification," in *Proc. of the Conf. on Evolutionary Computation*, Orlando, Florida, USA, 1994, pp. 120-124.

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

[3] D. Dutta, "Classification rules generation for iris data using lexicographic pareto based multi objective genetic algorithm," in *Proc. of 3rd Int. Conf. on Intelligent Systems and Networks*, Computational Intelligence Laboratory, Institute of Science and Technology, Klawad 133105, Harwana, India, 2009, pp. 173-177.

[4] M. V. Fidelis, H. S. Lopes, and A. A. Freitas, "Discovering comprehensible classification rules with a genetic algorithm," in *Proc. of the Congress on Evolutionary Computation*, San Diego, USA, 2000, pp. 805-810.

[5] A. A. Freitas, "A survey of evolutionary algorithms for data mining and knowledge discovery," Postgraduate Program in Computer Science, Pontificia Universidade Catolica do Parana, Rua Imaculada Conceicao, 1155, Curitiba - PR, 80215-901, Brazil, 2000.

[6] 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.

[7] A. Giordana, and F. Neri "Search-intensive concept induction," *Evolutionary Computation*, vol. 3, no. 4, pp. 375-416, 1995.

[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman, 1989.

[9] J. H. Holland, "Escaping brittleness: the possibilities of general-purpose learning algorithms applied to parallel rule-based systems," in *Machine Learning: An artificial intelligence approach*, vol. 2, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds. Los Altos, CA, USA: Morgan Kaufmann, 1986, pp. 593-623.

[10] M. Holsheimer, and A.P.J.M. Siebes "Data mining: The search for knowledge in databases," Amsterdam, CWI, Report CS-R9406, 1994.

[11] B. de-la. Iglesia, M. S. Philpott, A. J. Bagnall, and V. J. Rayward-Smith, "Data mining rules using multi-objective evolutionary algorithms," in *Proc. of the Congress on Evolutionary Computation*, Canberra, Australia, vol. 13, 2003, pp. 1552-1559.

[12] G. L. Pappa, A. A. Freitas, and C. A. A. Kaestner, "A Multiobjective Genetic Algorithm for Attribute Selection," in *Proc. of 4th Int. Conf. On Recent Advances in Soft Computing*, A. Lofti, J. Garibaldi, and R. John Eds. Nottingham Trent University, 2002, pp. 116-121.

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

[14] S. F. Smith, "A learning system based on genetic adaptive algorithms," PhD thesis, University of Pittsburgh, 1980.

[15] E. Zitzler, K. Deb, and L. Thiele "Comparison of multi-objective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173-195, 2000.

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 field of interest are Data Mining, Soft Computing etc.