Classification Rules Generation for Iris Data Using Lexicographic Pareto Based Multi Objective Genetic Algorithm

Dipankar Dutta*

* University Institute of Technology, Burdwan University/CSE & IT Department, Burdwan, West Bengal, India

***Abstract*—Extracting classification rules is one of the problems in data mining knowledge discovering process. Classification rule mining problems can be considered as a multi-objective problem rather than as a single objective one. Each chromosome represents a classification rule. We are using three measures c*onfidence, coverage* and *attractiveness* for evaluating a rule. These measures can be thought of as different objectives of Multi-Objective Genetic Algorithms. Using these three measures as the objectives of rule mining problem, this article uses a Lexicographic Pareto based Multi Objective Genetic Algorithm to extract some useful and attracting rules from Iris database.**

I. INTRODUCTION

This work present a system based on multi objective genetic algorithm (MOGA) to find out classification rule. The use of GAs in classification is an attempt to effectively exploit the large search space usually associated with classification tasks [9]. In essence, GAs is "search algorithms based on the mechanics of natural selection and natural genetics" [11]. It can be used in many fields. One of the advantages of GAs over "traditional" search methods is that the former performs a kind of 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 or local minima, in comparison with local search methods. It is attracting 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 a kind of local search [9]. In classification problems records are assigned to one out of a small set of pre-defined classes. Application domains include medical diagnosis, fault detection in electro-mechanical devices, evaluating credit applications, predicting crop quality, etc [10].

This paper is organized as follows: Section II describes some of the steps for data preprocessing. Section III states about Single Objective and Multi Objective Genetic Algorithm (MOGA), how MOGA can be used for classification rule mining, different approaches of doing so and fitness measures used in MOGA. Section IV presents the parameters value for Genetic Algorithms. Section V briefly describes different approach of encoding the chromosome and how we are doing that. Section VI describes how we are measuring accuracy of generated rules for classification. Section VII says about the database. Section VIII gives results. Finally section IX concludes the paper.

II. DATA PREPROCESSING

It include the following steps (among others) [1]:

A. *Data Integration*

This is necessary if the data to be mined comes 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.

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, etc.

C. *Discretization*

This step consists of transforming a continuous attribute into a categorical (or nominal) attribute, taking on only a few discrete values. We have calculated maximum and minimum value of an attribute to find the range of that attribute. Then we are dividing the range into 2,4,8 and 16 divisions so that if we convert data into binary 1,2,3 or 4 bits respectively will be suitable to store those data. In this way continuous attribute can be presented in binary notation. How many bits will be used that will depends on the range of values of the attributes.

D. *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 the filter approach the attribute selection method is independent of the data-mining algorithm to be applied to the selected attributes.

By contrast, in the wrapper approach the attribute selection method uses the result of the data mining algorithm to determine how good a given attribute subset is.

III. MULTI-OBJECTIVE GENETIC ALGORITHMS(MOGA)

A. *Single-Objective vs Multi-Objective problem*

In a single objective problem often we have to find out maximum or minimum value of one objective of a particular problem, which is the objective of that problem. There will be only one solution of these types of problem where we are getting minimum or maximum value of objective. Integer programming, dynamic programming,

geometric programming, stochastic programming and various other methods are available to solve these types of problems. Not enough emphasis is usually given to multi-objective optimization.

In multi objective problem objective is not one. There are many objectives that we have to optimize. In majority of the studies involving multiple objectives complexities involved in true multi-objective optimization is avoided and convert multiple objectives into a single objective function by defining some user defined parameters. We can say that single-objective optimization is a degenerate case of multi-objective optimization and multi-objective optimization is not a simple extension of single-objective optimization [3].

B. MOGA for Classification Rule Mining Problem

To cope with multi-objective problem like classification rule mining one can reviews three different approaches, namely: i) weighted sum approach ii) the lexicographical approach, where the objectives are ranked in order of priority and iii) the Pareto approach which consists of as many non-dominated solutions as possible and returning the set of Pareto front to the user. One can conclude that 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 lexicographic and the Pareto approach are more principled approaches and therefore deserve more attention from the data mining community. These approaches are discussed below.

C. Approaches for solving multi-objective problem

The three broad categories to cope with multi-objective problem is as follows:

Weighted sum approach: Transforming a multi-objective problem into a single objective problem by far the most commonly used approach in data mining literature. Normally, this can be done by a weighted sum of objective functions. That is the fitness value 'F' of a given candidate rule is typically measured by the formula: $F= w_1*f_1+w_2*f_2+\ldots+w_n*f_n$,where w_i , i=1,2…,n denotes the weight assigned to criteria f_i and n is the number of evaluation criteria. The strength of this method is its simplicity and ease of use. However, it has the drawbacks that, the setting of the weights in these formulas is ad-hoc, either based on a somewhat vague intuition of the user about the relative importance of different quality criteria or in trial and error experimentation with different weight values (which is mostly a difficult aspect of data mining). Hence the values of these weights can be determined empirically.

Another problem with these weights is that, once a formula with precise values of weights has been defined and given to a data-mining algorithm, the data-mining algorithm will be effectively trying to find the best rule for that particular settings of weights, missing the opportunity to find other rules that might be actually more attracting to the user, representing a better tradeoff between different quality criteria. In particular, weighted formulas involving a linear combination of different quality criteria have the limitation that they cannot find solutions in a concave region of the Pareto front.

Lexicographic approach: The basic idea of this approach is to assign different priorities to different objectives and then focus on optimizing the objectives in their order of priority. Hence, when two or more candidate rules are compared with each other to choose the best one, the first thing to do is to compare their performance measure for the highest priority objective. If one candidate rule is significantly better than the other with respect to that objective, the former is chosen. Otherwise the performance measure of the two candidate rules is compared with respect to the next highest objective. The process is repeated until one finds a clear winner or until one has used all the criteria. In the latter case, if there was no clear winner, one can simply select rules optimizing the highest priority objective. The lexicographic 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 lexicographic 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. In addition, although the lexicographic approach is somewhat more complex than the weighted-sum approach, the former can still be considered conceptually simple and easy to use.

Pareto approach: The basic idea of Pareto approach is that, instead of transforming a multi-objective problem into a single objective problem and then solving it by genetic algorithm, one should use multi-objective genetic algorithm directly. Adapt the algorithm to the problem being solved, rather than the other way around. In any case, this intuition needs to be presented in more formal terms, which is defined in the following.

Pareto dominance: A solution x1 is said to dominate a solution x_2 iff x_1 is strictly better than x2 with respect to at least one of the criteria (objectives) being optimized and x1 is not worse than x2 with respect to all the criteria being optimized. Using the Pareto dominance, solutions are compared against each other, i.e. a solution is dominant over another only if it has better performance in at least one criterion and non-inferior performance in all criteria. A solution is said to be Pareto optimal if it cannot be dominated by any other solution 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 solutions (Pareto set), one usually aims to derive a set of non-dominated solutions with objective values as close as possible to the objective values (Pareto front) of the Pareto set [15].

D. Classification rule mining

A classification rule is usually represented in the form of IF-THEN structure. IF<Some condition is satisfied> THEN <Predict class label>. The conditions associated in the IF part is termed as *Antecedent* and those with the THEN part is called the *Consequent*. In this article, we will refer them as A and C, respectively.

E. Fittness Measures

We are considering three measures for evaluating fitness of a classification rule as stated earlier. They are *Confidence, Coverage* and *Attractiveness*.

Confidence: It is the fraction of records, which satisfies all the conditions present in the rule i. e. antecedent part

and consequent class/no of records that satisfy antecedent part of the rule. This objective gives the accuracy of the rules extracted from the database.

Confidence = SUP(A&C)/SUP(A).

Where SUP(A) is the number of records satisfying all the conditions in the antecedent A.

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

Coverage: It is the fraction of records, which satisfies all the conditions present in the rule i. e. antecedent part and consequent class /no of records which satisfy consequent part of the rule. This measure is defined here as the proportion of the target class covered by the rule.

Coverage = SUP(A&C)/SUP(C).

Where SUP(C) is the number of records satisfying all the conditions in the consequent C.

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

Attractiveness: It measures how much attracting the rule is.

Attractiveness =1/(No of attribute in the antecedent part).

The generated rule may have a large number of attributes involved in the rule thereby making it difficult to understand [9]. If the generated rules are not understandable to the user, the user will never use them. Smaller rules are more attracting than larger rules.

For a particular set of antecedent attributes in a rule we are calculating support count for each class and we are assigning class label which class is giving highest support count. In other word chosen predicted class can be the class that has more representative in the set of examples satisfying the rule antecedent [14].

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

F. Elite Selection

As we are following Elitist MOGA we are selecting chromosomes (rules) based on their rank in the population produced after crossover and mutation operation of GA. Algorithm for selection is given below.

1) Initialized Rank=1;

2) Select rules for each class where confidence equals maximum confidence of that class from available rule set.

3) Among these rules select rules for each class where coverage equals maximum coverage of that class.

4) Among these rules select rules for each class where attractiveness equals maximum attractiveness of that class.

5) Among these rules we are selecting rule that is covering maximum range.

6) Apend the selected rule into selected rule set.

7) Rank rules as per value of the rank veriable.

8) Rank=Rank+1.

9) Delete all rules of that class if selected rule have confidence=1,coverage=1 and attractiveness=1 which are highest values of these objectives.

10) Otherwise delete rules, which are within the range of selected rule from available rule set.

11) Go to step 2 unless Rank>=User specified value

Here we are using a combination of lexicographic and pareto approach. Giving highest importance on the accuracy of the rules, rules with highest confidence are selected first. Then we are giving importance on coverage of the rules. Lowest importance is given to the attractiveness of the rules.

Getting true pareto front is very difficult. For example if we consider following rule set

Rule1:
Confidence=1,Coverage=0.99,Attractiveness=0.99

Rule2:
Confidence=0.99,Coverage=1,Attractiveness=0.99

Rule3:
Confidence=0.99,Coverage=0.99,Attractiveness=1

We will get maximum Confidence=1, maximum Coverage =1 and maximum Attractiveness =1. No rule will be selected as per dominance criteria for constructing pareto optimal front.

By applying our algorithm we can select Rule1, which is very close to the pareto optimal front.

IV. PARAMETERS FOR GENETIC ALGORITHMS FOR RULE GENERATION (GARG)

The parameters that were experimented with in this case were:

A. Crossover

Single point crossover with Crossover rate of 50% fixed.

B. Stopping condition

In this algorithm the stopping condition was implemented as a prefixed number of generations set by the user. At the end of each generation we are measuring the no of rules produced. We can also stop when we are not getting new rules in consecutive large number of generation.

C. Population size

Initially we are starting with 20 initial populations. We have also used an elitist reproduction strategy, where some top ranked rules of each generation was passed unaltered to the next generation. So population size is varying.

D. Mutation rate

10% fixed.

E. Selection

Before first run of GA 20 initial records are selected randomly from the population. Then we are choosing any number of attributes (may vary between 1 to 4) out of 4 attributes as antecedent.

For subsequent run uniform random selection was used to select a chromosome from the available set produced after elitist reproduction. This will reduce the selection pressure since roulette selection resulted in premature convergence (at the start of the run all rules performed very poorly as few rules matched any instance)[10]. Other

chromosomes are selected which are having 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 no of attributes selected as antecedent is varying.

V. Individual Representation And Encoding

A. *Pittsburgh Vs Michigan approach.*

Genetic algorithms for rule discovery can be divided into two broad approaches, based on how rules are encoded in the population of individuals ("chromosomes"). In the Pittsburgh approach each individual encodes a set of prediction rules. Ken De Jong and Steve Smith [13] from the University of Pittsburgh have popularized this.

In our work we have followed Michigan approach where each individual encodes a single prediction rule representing one class of Iris. This approach to building classifier systems was originally proposed by John Holland at the University of Michigan [12]. We are measuring accuracy of the rules by applying all rules on each record.

B. *Individual Encoding*

A chromosome may have n genes, where each gene corresponds one attribute, and n is the number of antecedent attributes in the data being mined. When an attribute is numeric it is represented by a set of Binary-coded lower and upper limits, where each limit is allocated a user-defined number of bits [2] (noOfBits). By default noOfBits=4. There is a scaling procedure that transforms any number in the range of possible values using noOfBits bits $[0,2^{noOfBits} - 1]$ to a number in the range of values that the attribute can take. So $noOfDivision=2^{noOfBits}$ .The procedure works as follows. When the data is loaded the maximum value (MaxValue) **,** and minimum value (MinValue), for each attribute stored. So each bit string representing a range of value an attribute calculated as rangeOfDivision=(MaxValue-MinValue)/ $2^{noOfBits}$ 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 upper bits with 0.

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

If attribute is categorical then no of different category=$2^{noOfBits}$.

The genes are positional, i.e. the first gene represents the first attribute, the second gene represents the second attribute, and so on as we are selecting attributes for a particular generation of GA. 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.

For each gene that is representing one attribute first part represent minimum value of the attribute and second part represent maximum value of the attribute in that rule.

In Iris dataset maximum value of n is 4. At a particular run of GA all 4 attributes may be chosen or some of the attribute may be chosen. These attributes may be any attributes out of these 4 attributes. Length of the chromosome = 2X(Summation of bit length of individual attribute). This length is same for all chromosomes for a particular run of GA but we may choose different attributes for different run of GA. So length of chromosome is not same for all run i.e. it is varying. Hence, different individuals correspond to rules with different number of predicting attributes. Since length of all chromosome at a particular ran of GA is same we can avoid complexity of designing GA operator such as crossover operator for handing variable length chromosome.

VI. Measuring Accuracy

For each record in the data set for each antecedent we are checking whether that antecedent value is within maximum and minimum value of a rule. We are choosing rule, with maximum confidence value. If more than one rule is giving same confidence we are choosing rule with higher coverage for classification. If more than one rule is giving same confidence and coverage we are choosing rule with higher attractiveness. In that manner we are classifying all data in a particular class 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 or it is misclassified.

Accuracy in %=Total no of matchX100/Total no of candidate

VII. Data Sets Used in the Experiments

We selected Iris Database [5] from large collection of test data at The UCI Repository of Machine Learning Databases [4]. The dataset consists of 50 samples from each of three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor) having class label 1,2 and 3 respectively. Four features were measured from each sample, they are the length and the width of sepal and petal.

VIII. Results

On testing Isis data with 150 training data and 150 testing data after 500 generation of GA we are getting 31 rules with 98% accuracy during one run. On average we are getting accuracy of 94%, which are compatible with other methods mentioned by Shyi-Ming Chena and Yao-De Fang in their paper [6] as describe in Table I and the method proposed by them.

IX. Conclusion

This paper illustrates classification rule mining method by the use of Multi Objective Genetic Algorithm (MOGA). A set of rules with high confidence, coverage and attractiveness value is generated. Repeating rules can be eliminated with the use of Range factor of the rules. Rules are giving high accuracy rate while predicting class label. In future algorithms can be improved to eliminate overlapping rules. This can be tested on different databases especially large-scale database. More elaborated experiments to optimize several parameters like crossover & mutation rate and rank is necessary. Paralleling of this algorithm can be done.

TABLE I.
A COMPARISON OF THE AVERAGE CLASSIFIACTION ACCURACY RATES FOR DIFFERENT CLASSIFICATION METHOD

Methods	Average Accuracy Classification Rates
Hong-and-Lee's method [16] (training data set: 75 instances; testing data set: 75 instances; executing 200 runs)	95.57%
Hong-and-Lee's method [17] (training data set: 75 instances; testing data set: 75 instances; executing 200 runs)	95.57%
Chang-and-Chen's method [18] (training data set: 75 instances; testing data set: 75 instances; executing 200 runs)	96.07%
Wu-and-Chen's method [19] (training data set: 75 instances; testing data set: 75 instances; after executing 200 runs)	96.21%
Chen-and-Fang's method [6] (training data set: 75 instances; testing data set: 75 instances; executing 200 times)	96.28%
Castro's method [20] (training data set: 120 instances; testing data set: 30 instances; executing 200 runs)	96.60%
Chen-and-Fang's method [6] (training data set: 120 instances; testing data set: 30 instances; executing 200 runs)	96.72%
Dasarathy's method [8] (training data set: 150 instances; testing data set: 150 instances)	94.67%
Hong-and-Chen's method [7] (training data set: 150 instances; testing data set: 150 instances)	96.67%
Chen-and-Fang's method [6] (training data set: 150 instances; testing data set: 150 instances)	97.33%

REFERENCES

[1] A. A. Freitas, "A survey of evolutionary algorithms for data mining and knowledge discovery," Pontilicia Universidade Catolica do Parana Rna Imaculada Conceicao, Brazil, (Year), unpublished.

[2] B. de-la. Iglesia, M. S. Philpott, A. J. Bagnall, and V. J. Rayward-Smith, "Data mining rules using multi-objective evolutionary algorithms," *Proceedings of the Congress on Evolutionary Computation,* vol. 3, pp. 1552 – 1559, 2003.

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

[4] http://www.ics.uci.edu/-learn/MLRepository.html.

[5] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics,* vol. 7, pp.179 – 188, 1936.

[6] S. M. Chen, and Y. D. Fang, "A new approach for handling the iris data classification problem," *International Journal of Applied Science and Engineering*, vol. 3, no. 1, pp. 37 - 49, 2005.

[7] T. P. Hong, and J. B. Chen, "Finding relevant attributes and membership functions," *Fuzzy Sets and Systems,* vol. 103, pp. 389 - 404, 1999.

[8] B. V. Dasarathy, "Noise around the neighborhood: A new system structure and classification rule for recognition in partially exposed environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, pp. 67 - 71, 1980.

[9] M. V. Fidelis, H. S. Lopes, and A. A. Freitas, "Discovering Comprehensible Classification Rules with a Genetic Algorithm," *Proceeding of the Congress on Evolutionary Computation,* CA, USA, pp. 805 – 810, 2000.

[10] A. L. Corcorau, and S. Sen, "Using real-valued genetic: Algorithms to evolve rule sets for classification," *IEEE-CEC,* pp 120 – 124, 1994.

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

[12] J. H. Holland, "Escaping brittleness: the possibilities of general-purpose learning algorithms applied to parallel rule-based systems," In R. Michalski, J. Carbonell, and T. M. Mitchell, editors. *Machine Learning, an artificial intelligence approach.* Morgan Kaufmann, Los Alamos, CA, vol. 2, 1986.

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

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

[15] Alex A. Freitus, "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.

[16] T. P. Hong, and C. Y. Lee, "Induction of fuzzy rules and membership functions from training examples," *Fuzzy Sets and Systems*, vol. 84, pp. 33 - 47, 1996.

[17] T. P. Hong, and C. Y. Lee, "Effect of merging order on performance of fuzzy induction," *Intelligent Data Analysis*, vol. 3, pp. 139 - 151, 1999.

[18] C. H. Chang, and S. M. Chen, "Constructing membership functions and generating weighted fuzzy rules from training data," *Proceedings of the Ninth National Conference on Fuzzy Theory and Its Applications*, Taiwan, pp. 708 - 713, 2001.

[19] T. P. Wu, and S. M. Chen, "A new method for constructing membership functions and fuzzy rules from training examples," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 29, pp. 25 - 40, 1999.

[20] J. L. Castro, J. J. Castro-Schez, and J. M. Zurita, "Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems," *Fuzzy Sets and Systems*, vol. 101, pp. 331 - 342, 1999.