MMDT: Multi-Objective Memetic Rule Learning from Decision Tree

Bahareh Shaabani [a*], Hedieh Sajedi [b]

[a] Faculty of Computer and Information Technology Engineering,Qazvin Branch,Islamic Azad University,Qazvin,Iran
[b] Assistant Professor,Department of Computer Science, Tehran University, Tehran,Iran

Received 20 July 2012; accepted 15 May 2013

Abstract

In this article, a Multi-Objective Memetic Algorithm (MA) for rule learning is proposed. Prediction accuracy and interpretation are two measures that conflict with each other. In this approach, we consider accuracy and interpretation of rules sets. Additionally, individual classifiers face other problems such as huge sizes, high dimensionality and imbalance classes' distribution data sets. This article proposed a way to handle imbalance classes' distribution. We introduce Multi-Objective Memetic Rule Learning from Decision Tree (MMDT). This approach partially solves the problem of class imbalance. Moreover, a MA is proposed for refining rule extracted by decision tree. In this algorithm, a Particle Swarm Optimization (PSO) is used in MA. In refinement step, the aim is to increase the accuracy and ability to interpret. MMDT has been compared with PART, C4.5 and DTGA on numbers of data sets from UCI based on accuracy and interpretation measures. Results show MMDT offers improvement in many cases.

Keywords: C4.5, Memetic Algorithm, rule sets, Particle Swarm Optimization

1. Introduction

In the last decades, an increasing research interest in the fields of data mining and knowledge discovery is observed [1]. Data Mining (DM) is the process of knowledge discovery, which searches a large volume of data to discover attractive and useful information previously unknown [2]. Recently, the amount of data stored in databases is growing fast. This large amount of stored data includes important hidden knowledge and information, which could be used to develop the decision-making process of an organization, fraud detection, and customer retention, to make control and science exploration [3]. For instance, data about before bank loan might include interesting relationships between loan and customers. The discovery of such relationships can be very useful to recognize the loyal customers. However, the number of human data analysts grows at a much smaller rate than the amount of stored data. So, there is clear need for (semi-) automated methods for extracting knowledge from data. Classification is a form of data analysis that extracts models describing important data classes. Such models, called classifiers, predict categorical (discrete, unordered) class labels. It produces from a set of training examples a set of rules to classify future test data. Rule induction is one of the most common forms of knowledge discovery. Different criteria exist to evaluate classification algorithm performance including prediction accuracy, interpretation (interpretability) rule and scalability. Some of these criteria conflict with each other. Also different algorithms were proposedin classification task. Each of them uses different measures and is applied on different data sets. So, different algorithms have different advantages and disadvantages. But, few algorithms have been proposed to consider all these measures. Different algorithms have been proposed with different criteria. A summary of these classification tasks are proposed as follows:

Over the years, GAs have been successfully applied in learning tasks in different domains like chemical process control [3], financial classification [4], manufacturing scheduling [5], robot control [6], etc. A population of a fuzzy rule set [7] was evolved using a Genetic Program (GP: an extension of a GA) [8]. The used metric in this approach was prediction accuracy. An accuracy based GA approach, UCS [9], was developed for performing the classification task. C4.5 [10] is one of the most successful and popular rule induction algorithms. In order to predict the future sales of a printed circuit board factory more precisely, the research in [11] proposed a hybrid model in which a GA was used to optimize the Fuzzy Rule Base (FRB) accepted by the Self-Organization Map (SOM) Neural Network with two metric prediction accuracy and average number of conditions. The GA part of the hybrid

* Corresponding author. Email: Bahareh.shabany@gmail.com

model was employed to find an optimal structuring element for classifying garment defect types in [12]. Faraoun and Boukelif made an attempt to show the use of a new GP classification approach for performing network intrusion detection in [13]. The research in [14] proposed a decision support tool, combining an expert system and the Takagi–Sugeno Fuzzy Neural Network (TSFNN) for fashion coordination. They have also shown that the GA plays an important role in reducing the number of coordination rules and the training time for TSFNN. The research in [15] proposed a hybrid model to extract accuracy based rule sets. The research in [16] considered the induction of fuzzy classification rules for data mining purposes and suggested a hybrid genetic algorithm for learning approximate fuzzy rules. The research in [17] proposed an elitist multi-objective genetic algorithm (EMOGA) for mining classification rules from large databases and emphasizes on predictive accuracy, interpretation and interestingness of the rules. A hybrid GA and fuzzy logic is proposed for extracting linguistic rules from data sets. The research in [18]formalized linguistic rules based on complex linguistic data summaries, in which the degree of confidence of linguistic rules from a data set can be explained by linguistic quantifiers and its linguistic truth from the fuzzy logical point of view. In order to obtain a linguistic rule with a higher degree of linguistic truth, a genetic algorithm was used to optimize the number and parameters of membership functions of linguistic values. The research in [19] proposed a novel Genetic Swarm Algorithm (GSA) for obtaining near optimal rule set and membership function tuning. Advanced and problem specific genetic operators were proposed to improve the convergence of GSA and classification accuracy. The research in [20] introduces an accuracy-based learning system called DTGA (decision tree and genetic algorithm) that aims to improve prediction accuracy over any classification problem irrespective to domain, size, dimensionality, and class distribution. A rule-based knowledge discovery model, combining C4.5 (a Decision Tree based rule inductive algorithm) and a new parallel genetic algorithm based on the idea of massive parallelism, was introduced in [21]. The prime goal of the model was to produce a compact set of informative rules from any kind of classification problem. An Evolutionary Memetic Algorithm for rule extraction was proposed in [22] which uses a micro-Genetic Algorithm based (μGA) technique, and EMA-AIS, which is inspired by Artificial Immune System (AIS) and uses the clonal selection for cell proliferation. The metric is used in this paper is accuracy. An evolutionary stratified training set selection for extracting classification rule with trade off precision-interpretability was proposed in [23]. Also, this method faces scaling problem that appears in the evolution of large data sets. In this paper, a new training set selection was suggested for large size sets. Another approach of multi-objective genetic algorithm that could consider the accuracy, interpretation and definability of approximate rule was expressed in [24].

Particle swarm optimizer (PSO) was another evolutionary algorithm, which simulated the coordinated movement in flocks of birds. The research in [25] proposed the use of PSO for data mining. PSO can achieve the rule discovery process. The rule representation in PSO used the Michigan approach. PSO needs fewer particles than GA to achieve the same results. The research in [26] proposed an algorithm for generating fuzzy rules. This algorithm is mainly based on both concepts of data mining and PSO algorithm. The research in [27] proposed a new way for rule discovery as a multi objective optimization problem with to criteria, predictive accuracy and ability to interpret. A multi-objective PSO algorithm was proposed in [27] to solve the problem. A new Discrete Particle Swarm Optimization approach to induce rules from the discrete data was proposed in [28]. The algorithm initializes its population by taking into account the discrete nature of the data. It assigns different fixed probabilities to current, local best and the global best positions. One of the important problems in the design of fuzzy classifiers is the formation of fuzzy if-then rules and the membership functions. The research in [29] considered a hybrid Particle Swarm Optimization based approach for fuzzy classifier design which incorporates the concept of mutation from evolutionary computations. A hybrid PSO/ACO algorithm for discovering classification rules is proposed in [30]. In PSO/ACO, the rule discovery process is divided into two separate phases. In the first phase, ACO discovers a rule included nominal attributes only. In the second phase, PSO discovers the rule potentially extended with continuous attributes.

In this paper, we propose a new approach using C4.5 and MA. C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan [10] and an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. In this paper, C4.5 is used for rules induction. And then “if ...then” parts are removed from rules due to they are not suitable for applying in memetic algorithm. MMDT is tested on data set obtained from UCI repository [31]. Continuous attributes are discretized by Yet. Another Boosting Approach for C4.5 Algorithm (YABAC4.5) algorithm is [32].This approach tries to retain best rules. In this article, the learning capabilities of C4.5 and MA are combined to improve the performance of the classification problems. In MMDT, we tend to obtain rule sets with high precision and ability to interpret.

There are some restrictions in using some of the traditional machine learning methods for data mining. One of the biggest restrictions is the problem of scaling up the methods to handle the huge size of the data sets and their high dimensionality. Imbalanced data sets have significantly unequal distributions between classes. The between-class imbalance causes conventional classification methods to favour majority classes, resulting in very low or even no detection of minority classes. Imbalanced data sets exist in many real-world applications, where the sizes of majority classes severely exceed those of the minor classes.

For example, in international patent classification, some major classes have up to hundreds of thousands of samples while some minor classes have less than ten samples. A fundamental issue of learning from imbalanced data sets is serious performance degradation of standard learning algorithms, such as back-propagation algorithm. Most standard algorithms assume or expect balanced class distributions or equal misclassification costs. Therefore, when serious imbalanced data sets are presented, these algorithms fail to properly represent the distribution characteristics of the data and provide predictions favourable to the majority classes[33].

Evolutionary Algorithms (EAs) have the ability of escaping local optima due to their inherent global search capability. Their concurrent search enables them to promptly explore and identify new promising regions of the solution space. Although EAs are able to see the macro situation well, they do not exploit the search space thoroughly. Hence, local search is often used as a complement to EAs optimization that concentrate mainly on global exploration. So, we use a hybrid MA and C4.5 algorithm for classification task. In the past, many researches emphasized a special type of problem, and some of them are designed to solve multi-object problem such as [17, 27]. We put emphasis on prediction accuracy and the ability to interpret. In this paper, a hybrid GA and PSO is introduced for refining C4.5 rule sets. Also, a new splitting technique is proposed for tackling imbalance class problems. In this article, we aim at satisfying the classification criteria of high accuracy and ease of user comprehension [34].In practice, the interpretation measure is a kind of subject concept as it varies from user to user. However, the data mining journalism uses an objective measure: generally, the smaller the rule, the more comprehensible it is. There are various ways to measure rule interpretation [35–38]. The standard way of measuring interpretation is to count the number of rules and the number of conditions in these rules [39].

Our hybrid approach is so easy to implement: C4.5 algorithm produces rule set and these rules are encoded to a form that can be applied in MA. We use MA to refine and improve rule sets in huge sizes and high dimensionality data sets until we obtain high accuracy and ability to interpret. The proposed approach is tested on six datasets from UCI. The experimental result shows the accuracy of MMDT in all cases is comparable with other approaches. The rule sets and condition in each rule obtained from MMDT are less than the other approaches. In many cases, the interpretation the rules obtained from MMDT is better than C4.5 and PART.

This paper is organized as follows: Our proposed approach is discussed in Section 2. Section 3 describes experimental design and analysis. Finally, Section 4 concludes this article.

2. Proposed Approach: MMDT

In this section, we will have an overview of the system. A dynamic splitting technique is introduced to reduce imbalanced problem of data sets. Then, we present a short review of C4.5. A review of memetic algorithm, its encoding and decoding strategy and operators is proposed. Finally, MMDT is introduced in details. This is a multi-objective memetic algorithm. So, we emphasize predictive accuracy and interpretation of the achieved rules. Empirical results of MMDT are compared with C4.5, PART and DTGA algorithm. Fig. 1 shows the conceptual model of the proposed learning system. Also, the procedures involved in the phases are discussed in this section. However, before discussing them in details, we must first describe the proposed sampling strategy adopted in the current model.

2.1. YABAC4.5

This paper continues attributes discreted by YABAC4.5. Additionally, the attributes with missing value are handled. This is our previous work on disctretization.

2.2. A Dynamic Data Splitting Method

Recently, the imbalanced data-set problem has required more attention in the field of machine learning research [19]. This problem happens when the number of samples of one class is much lower than the samples of the other classes. This problem is so important. Most classifiers generally perform weakly on imbalanced data-sets because they are planned to reduce the global error rate, and in this way they tend to classify almost all instances as negative (i.e., the majority class). But minority class maybe most important classes for example in fraud detection, cancer diagnosis, network influence and so on. It has been proved that applying a pre-processing step in order to balance the class distribution is a positive solution to the problem of imbalanced data-sets [40]. Different sampling methods are classified into three groups:

- Under-sampling methods that create a subset of the original data-set by eliminating some of the examples of the majority class.
- Over-sampling methods that create a superset of the original data-set by replicating some of the examples of the minority class or creating new ones from the original minority class instances.
- Hybrid methods that combine the two previous methods, eliminating some of the minority class example expanded by the over-sampling method in order to eliminate overfitting [39].

Although, each of the above groups has some weakness such as computational load raises due to over-sampling, all existing training data are not taken into account in under-sampling. In fact, there is no good solution for such a problem [20].

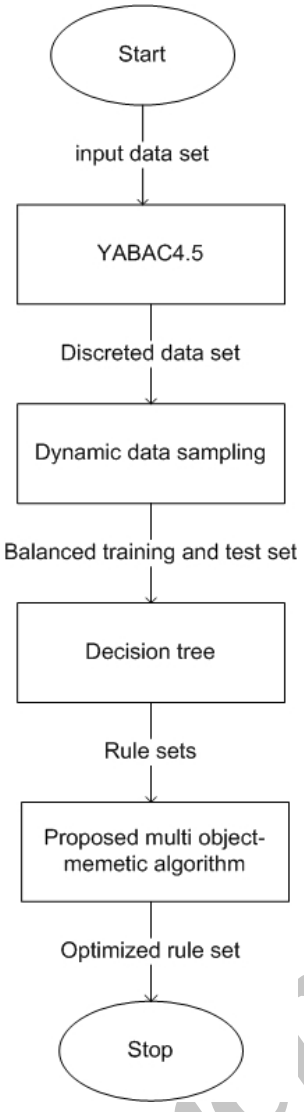

Fig.1: Steps of MMDT

In this paper, we propose a new dynamic sampling method to select training data. This dynamic method tries to reduce imbalance of training data. Our dynamic technique performs in this way: In this method, about 30% samples of majority class (es) is(are) selected for training set. But other class samples can be selected up to70% for training sets. Other remaining samples are chosen for test set. By using this method, we tend to reduce the imbalance problem.

In this approach, we want to reduce the impact of imbalance problem. For example, if there are 5 class values (c1, c2, c3, c4, c5) in classification problem S with 170 samples in total, and number of samples of classes: c1, c2, c3, c4, c5 are 30, 50, 50, 25, 15, respectively. Majority class is c2 and c3, so 30% of samples from majority classes are selected. And other class needs to consider in above algorithm. Then, 15, 15, 15, 15, 11 samples of class-types c1, c2, c3, c4, c5, respectively, are chosen. Dynamic sampling algorithm is as follow:

Variables:
μ : number of majority class;
ω : number of samples;
η: percentage of sample for selecting;
Input: imbalance data set

For (all classes do)
  If (($\omega == \mu$)
    Randomly select 30% of samples;
  Else
    If ($\mu / \omega <=2$)
      Randomly select $\eta\% = ceil(\omega * 0.3 + \mu / \omega)$;
    Else
      Randomly select 70% sample;
    End
  End
End

Output: balance data set

2.3. Decision Tree

Decision trees and rule induction algorithms are very important techniques and they are used generally in DM [41]. They are able to make human-readable descriptions of trends in the underlying relationships of a data set and can be used for classification and prediction tasks. Advantages of decision tree include inexpensive construction, being extremely fast at classifying unknown records, being easy to interpret for small-sized trees, and being comparable accuracy to other classification techniques for many simple data sets[20].

During the late 1970s and early 1980s, J. Ross Quinlan developed a decision tree algorithm known as ID3. This work extended the earlier work on concept learning systems, described by E. B. Hunt, J. Marin, and P. T. Stone. Quinlan later presented C4.5 which became a benchmark to which newer supervised learning algorithms are often compared. ID3and C4.5 accept a greedy (i.e., non backtracking) approach in which decision trees are constructed in a top-down recursive divide-and-conquer manner. Most algorithms for decision tree induction also follow a top-down approach, which starts with a training set of tuples and their associated class labels. The training set is recursively partitioned into smaller subsets as the tree is being built.C4.5 is an algorithm used to generate a decision tree. The decision trees generated by C4.5 can be used for classification, and, for this reason, C4.5 is often referred to as a statistical classifier[42]. C4.5 builds decision trees from a set of training data using the concept of information entropy:

$$Entropy(S) = \sum -P_i \log P_i \quad (1)$$

where p_i is the proportion of S (the collection of examples) belonging to class i out of c (number of) classes.

2.4. Proposed Memetic Algorithm

Memetic algorithms (MAs) are evolutionary algorithms (EAs) that apply a separate local search process to refine the individuals (i.e. improve their fitness byhill climbing, etc.). MAs are inspired by Richard Dawkins concept of a meme which represents a unit of cultural evolution that can exhibit local refinements. They are combined with some kinds of local search and are able to balance the exploration and exploitation capabilities of both genetic algorithm and local search [42].In this section, memetic algorithm encoding strategy, operators and fitness measure are presented. Then, the proposed approach is introduced.

2.4.1. Encoding and Decoding Strategy

In this phase of our proposed system, the discretized values are directly used in C4.5 algorithm. Then each attribute in rules is converted to six bits to be use in memetic algorithm. In this six-bit binary encoding, each discrete decimal value is simply represented by its equivalent binary number, except the '*' which is represented by all 0's. On every block of six bits of binary rule, decoding is performed using the reverse of the above mentioned encoding scheme to be converted again to the equivalent decimal value.

2.4.2. Fitness Function

We use a fitness function that has been used in [20]:

$$f(r_{acc(i)}) = \frac{m-n}{t} \qquad (2)$$

where $r_{acc(i)}$ represents accuracy of i-th rule, m is the number of training examples satisfying all the conditions in the antecedent (A) and the consequent (C) of the rule (r_i) too, n is the number of training examples which satisfy all the conditions in the antecedent (A) part but not the consequent (C) of the rule (r_i) and t is the total number of training examples. This fitness function tries to retain the best weighted rule minimizing the value of n (i.e., classification error), and reduces the chances of the same fitness value occurring among the rules [20].In addition to this fitness function, we use of condition in each rule for comparing:

$$f(r_{com(i)}) = L \qquad (3)$$

Where $r_{com(i)}$ represents interpretation of i-th rule, L is number of condition in each rule. We want to reduce number of conditions to rule to increase ability to interpret. Each rule with better accuracy and interpretation replace with worst rule.

Further, the approach described here is aware of the overall fitness of the new rule set, i.e., if the overall fitness of the new rule set (replacing the worst rule of the old set by the new offspring) increases, only then the new one is accepted. The overall fitness function is defined as follow:

$$F(R) = \frac{N}{T} \times 100 \qquad (4)$$

where N is number of test examples covered by the rule set and T is total number of test example.

2.4.3. Selection and Crosover

In this paper, a single-point crossover site is considered for two classifiers randomly chosen from the population (the set of rules).Also, the point is chosen randomly within the length of the classifier.

2.4.4. Local Search

The global search ability of the evolutionary part of a memetic algorithm takes care of exploration, trying to identify the most hopeful search space regions; the local search part examines the surroundings of some initial solution, exploiting it in this way. The role of the local search is fundamental and the selection of its search ruleand its harmonization within the global search schemes make the global algorithmic success of memetic frameworks. Many researchers complemented the global exploration capability of EAs by incorporating dedicated learning or local search heuristics. Experimental studies have shown that EA-LS hybrids or memetic EAs are capable of more efficient search capabilities [43]. In this paper, we use PSO as a local optimizer for refining rule set. PSO is a stochastic optimization method where a population of individuals (particles) moves through the search space. The rules, which govern the movement of particles, are inspired by the social interaction among a school of fishes or a flock of birds in nature. In a PSO model, a particle can be represented by its position and its velocity. At every iteration, each particle in the population can complete its updating based on its current velocity and position, the best position found so far by itself, and the best position found so far by any of its neighbours, which can be described as follows:

Let $X_i(t)$ represent the position of i-th particle in search space at time step t. The position of each particle is updated according to equation 5.

$$X_i(t+1) = X_i(t) + V_i(t+1) \qquad (4)$$

Where $V_i(t+1)$ is the velocity of particle i at time step t+1, and it is calculated as follows:

$$V_{i,j}(t+1) = C(W.V_{i,j}(t) + C_1 r_{1,j}(t)(X_{p,j}(t) - X_{i,j}(t)) + C_2 r_{2,j}(t)(ones(popsize,1)\, X_{g,j}(t) - X_{i,j}))) \qquad (5)$$

Where $V_{i,j}(t)$ denotes the j-th component of the i-th particles velocity vector at time step t; W is inertia weight index; C is constriction factor; $X_{i,j}(t)$ represents the jth

component of the ith particles position vector at time step t; C_1 and C_2 are positive acceleration constants used to scale the contribution of the cognitive and social components, respectively. And also $r_{1,j}$(t) and $r_{2,j}$(t) are uniformly distributed random values in [0, M], that M is number of features, $X_{p,j}$(t) is the best position visited by ith particle since the first time step. And finally $X_{g,j}$(t) is the best position found by swarm i.e. all particles. Both $X_{p,j}$(t) and $X_{g,j}$ (t) are determined by the use of a fitness function which evaluate each particle to find how close the corresponding solution is to the optimum. It is obvious that velocity vector drives the optimization process, and reflects both the Social and cognitive knowledge of particles. Originally, globalbest and localbest PSO algorithms have been developed which differ in the size of their neighborhoods [5]. In globalbest PSO, each particle is supposed to be the neighbor of all other particles. In localbest PSO the degree of connectivity among the population is less than the globalbest PSO.

Based on the approach of choosing globalbest, PSO can be classified into two versions, global and local. In the global version of PSO algorithms, all particles in the population ''share'' the same globalbest (the best fitness solution found so far by them). As a result, the population can converge quickly into one optimum in the search space. On the contrast, the local version of PSO only allows each particle to choose its globalbest from its neighbours, which only comprise part of the whole population, in a given distance space. Therefore, the particles in the population may converge into multiple different optima eventually in the local PSO model [44].

We apply PSO as a local optimizer for refining rule set. In this local search, we apply the following function as an objective function.

$$OF(r_i) = \frac{m-n}{t} \qquad (6)$$

This objective function is used as an accuracy fitness function in MA algorithm, where represents accuracy of i-th rule, m is the number of r_i training examples satisfying all the conditions in the antecedent (A) and the consequent (C) of the rule (r_i) too, n is the number of training examples which satisfy all the conditions in the antecedent (A) part but not the consequent (C) of the rule (r_i) and t is the total number of training examples.

2.5. MMDT as Rule Induction Algorithm

In this research, we intend to make a rule set with high accuracy and easy to interpret. Therefore after extracting rule from C4.5 algorithm, MMDT refines rule sets with memetic algorithm. In MMDT, first, all data sets discrete by YABAC4.5, then training set chooses according to the dynamic sampling method that proposed in Section 2.2, next, C4.5 is applied on two distinct data sets for train and test. Rule set with minimum number of rules is selected to refine. Finally, memetic algorithm refines rule set. In refining stage, we try to increase accuracy and decrease condition of rules. All the examples of a problem have the same number of attributes. Rules are in the form like 1 * 2 1 * 3 0 in which '*' is treated as the don't care value. Overall fitness of the generated rule set calculates from Eq. (4). The steps of the proposed approach are as the following:

Variables:

Maxgen: the maximum number of generations of the new optimized rule set;
gen: a new generation;
E_{train} : A set for training examples;
E_{test}: A set for test examples;
R: a set in which to store rules generated by any rule inductive algorithm;
R_T : a set in which to store a rule set for computing its accuracy;
O_1, O_2 : offspring;
P_1, P_2 : parents;
r_w: to denote the lowest fitness score rule;

Input:

R: rule set discovered by C4.5 from E_{train} ;
E_{train} : a set of training examples from which R is generated;
Fitness score (f (r_i)) of each rule (r_i) in R;
E_{test}: a set of test examples on which the overall fitness (F(R)) of the rule set R is to be computed;

Begin

gen←0;

Step-1:

Randomly select two parents: P1 and P2 from R. Values of attributes of these parents (rules) are encoded into related binary values;

Step-2:

Select accidentally a single-point crossover site within the length of the classifier. And Apply crossover on two parents P1and P2 at the crossover site;

Step-3:

Apply local search on two offsprings;

Step-4:

Step-4.1:

$O_1$and O_2 are decoded into decimal form;

If the valid O_1 is a second copy of any existing rule in R **then**

discards it and go to step-4.2;

else

compute f ($r_{acc(i)}$) and f ($r_{com(i)}$) of O1;

Find the rule r_w from R and compare f ($r_{acc(i)}$) and f($r_{com(i)}$) with the f ($r_{acc(i)}$) and f ($r_{com(i)}$) of O_1;

If the fitness of O_1 is lower and number of condition is more than that of r_w **then**

copy O_1 in place of r_w in R;

else

ignore O_1;

Step-4.2:

If the valid O_2 is a second copy of any rule in R **then**

discards it and go to step-6;

else

compute the f ($r_{acc(i)}$) and f ($r_{com(i)}$) of O_2 ;

Next, find the rule r_w from R and compare its f ($r_{acc(i)}$) and f ($r_{com(i)}$) with f ($r_{acc(i)}$) and f ($r_{com(i)}$) of O_2 ;

If the fitness value of O_2 is lower and number of condition is more than the fitness of r_w **then**

go to step-5;

else

compute the overall fitness (F(R)) of R_T from E_{test} (copying the current content of R into R_T and putting this new offspring (O_2) in place of r_w);

If the overall fitness of R_T is greater than the overall fitness of R **then**

copy O_2 in place of r_w in R;

else

discard O_2 ;

Step-5:

gen ← gen + 1;

If the desired number of generations is not completed (i.e., gen < Maxgen) **then**

go to step-1;

End

Output: Optimized rule set.

In MMDT, we intend to reduce the number of conditions in selected rules and increase accuracy of rules. In fact, we consider trade off between accuracy and ability to interpret. For this purpose, we presented a new hybrid algorithm for refining rule sets.

3. Experimental Results

In this section, we provide experimental result of our learning algorithm over six datasets. The algorithm istested on six benchmark data sets of realworld problems drawn from UCI machine learning repository. Table 1 shows relevant feature of these data sets. In this article, YABAC4.5discretizer converts each original data set into discrete form and handles missing values suitably and reduces number of attributes. The C4.5 is run on the three different combinations of rule set that separates by dynamic splitting algorithm to produce three different rule sets, and the accuracy and number of each rule set is computed on its respective test set. Now, the rule set with the minimum number of rule set is selected as the initial best rule set for applying the proposed algorithm. Parents and crossover sites of the MA are selected randomly, with a different random seed each time.

From Table 1, it is somewhat clear that the selected data sets are chosen from different domains. Again, these are very varied in terms of number of classes, number of features and number of instances. The number of classes ranges up to 19, the number of features ranges from 4 to 38 and the number of instances ranges from 123 to 648.

3.1. Experiment of Hybrid Learning Algorithm by Accuracy Measure

To have a good estimate, all the classifiers are run 10 times, each time on a distinct training set and a test set of each of six data sets. Note that each training set and test set against each data set are selected by our proposed data splitting strategy discussed in Section 2.1. Next, each classifier is trained on the training set, and then the trained model is run on the test set to measure accuracy and conditions in all rules. However, the training and test sets for each data set decided for each distinct run are used by all the classifiers for that run only. In other words, at every run, two distinct sets (one training set and one test set) for each data set are first decided following the suggested data sampling approach, and then individual classifier is trained and the induced knowledge is tested. Note that 30 generations are produced in each run of MA. The train accuracies and condition in each rule set on each data set achieved by individual classifier are averaged over all 10 results. In addition, a standard deviation along with each mean result is reported. Standard deviation is important, since it generalizes the overall performance of classifier. This algorithm compares with PART and C4.5 and DTGA. Accuracy of our method is calculated as follow:

$$F(R) = \frac{N}{T} \times 100 \qquad (7)$$

where N is number of test examples covered by the rule set and T is total number of test example.

In table 2, we show accuracy of our method and compare it with three other methods. Experiment results show this algorithm is more accurate than other algorithm. Fig. 2 shows clearly difference between MMDT and other

approaches. In next table, table 3, this approach is compare with several methods by ability to interpret.

Table 1: Characteristic of Datasets

Problem name	Number of attributes	Number of classes	Number of examples	% of minority class	% of majority class
Glass	9	6	213	4.2	35.1
Iris	4	3	150	33.3	33.3
Heart(h)	13	5	294	5.1	63.4
Pima	8	2	768	34.9	65.1
Heart(S)	12	5	123	4.06	39.2
Liver	6	2	345	42.2	57.8
Soybean	35	19	684	0.32	13.2

Table 2: Performance comparison of PART, C4.5, DTGA and MMDT with accuracy measure

Problem name	PART	C4.5	DTGA	MMDT
Glass	73.15±4.13	73.50±4.00	79.37±4.59	81.44±6.67
Iris	94.67±2.57	96.67±2.01	98.02±1.82	98.29±0.40
Heart(h)	72.79±2.35	72.65±1.74	78.08±4.63	86.58±2.55
Heart(S)	50.38±4.51	50.91±5.00	52.82±3.62	62.11±7.41
Liver disorder	83.65±2.84	84.83±4.39	80.02±1.85	88.20±6.78
Soybean	89.71±1.42	88.14±1.81	91.47±2.58	92.10±3.11

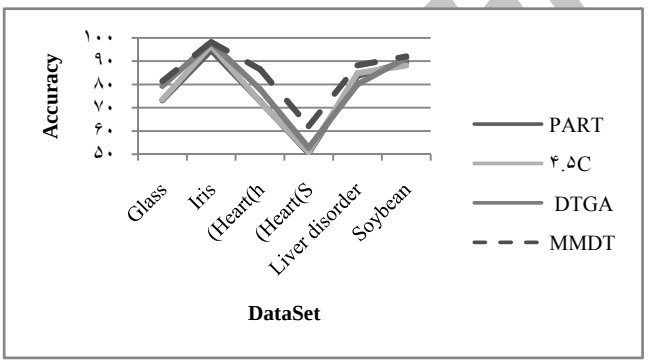

Figure 2: Performance comparison of PART, C4.5, DTGA and MMDT with accuracy measure

3.2. Experiment of Hybrid Learning Algorithm Interpretation Measure

In table 2, we evaluated the accuracy of our algorithm using six data sets. In table 3, we intend to compare the interpretation of our algorithm with several other algorithms.

Experimental results show that the accuracy of the MMDT is improved compared with other algorithms, so performance is better in terms of accuracy. While comparing the proposed algorithm with algorithms PART demonstrates the interpretation PART algorithm is better in two cases but in other case, MMDT is more efficient. Difference is shown clearly in fig. 3 and fig. 4.

Table 3: Performance comparison of PART, C4.5, DTGA and MMDT with interpretation measure

Problem name	PART	C4.5	DTGA	MMDT
Glass				
Average No. of rules	**11.55**	16.75	18	13.27
Number of condition in rule set	**25**	76	85	65
Iris				
Average No. of rules	**3**	5	6	5
Number of condition in rule set	**3**	9	9	6
Heart(h)				
Average No. of rules	10	9.30	12.3	**9**
Number of condition in rule set	50	47	56	**41**
Heart(S)				
Average No. of rules	8.15	9	10	**7**
Number of condition in rule set	36	73	30	**24**
Liver disorder				
Average No. of rules	**21.35**	45	43	24.8
Number of condition in rule set	78	140	137	**64**
Soybean				
Average No. of rules	40.37	50.85	48.5	**36.3**
Number of condition in rule set	127	194	186	**123**

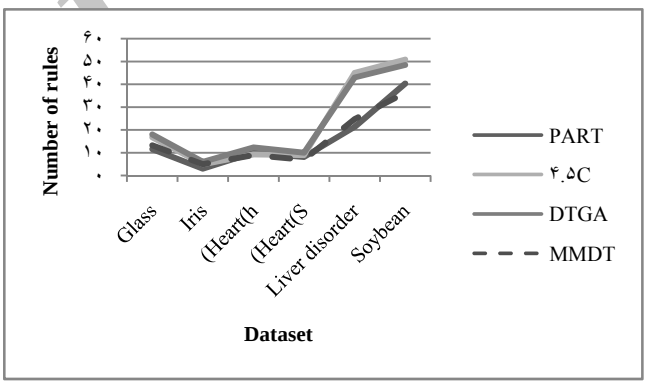

Figure 3: Number of rules with rule based algorithms

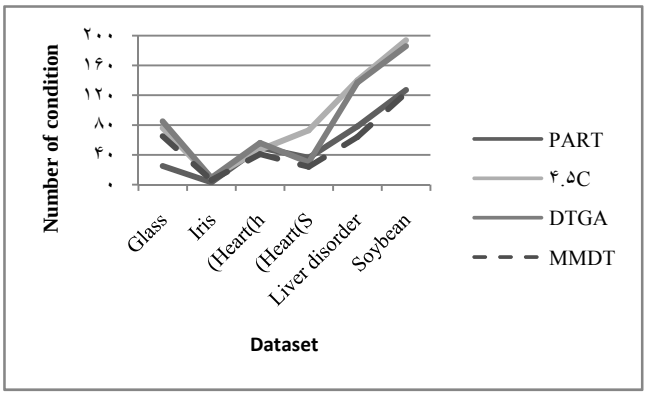

Figure 4: Number of condition with rule based algorithms

4. Conclusion

Different algorithms have been proposed for classification tasks each using different measures. These algorithms are used on different datasets. But, few algorithms have been proposed to deal with all these measures and all data sets. Moreover, individual classifiers have problems when dealing with huge sizes, high dimensionality and imbalance classes' distribution data sets.

In this paper, a hybrid multi-objective algorithm was proposed to extract rules. In this approach, first a decision tree was used for producing rules sets. Then MA was applied for refining rules. Two criteria for evaluating rules were accuracy and interpretation. These two measures conflict with each other. However, we tried to improve the two measures. A new dynamic splitting technique was proposed for imbalance class problems. MMDT handled the huge size of the data sets and their high dimensionality. This approach also tried to reduce the number of rule sets and conditions in each rule. Experimental results showed our multi-objective algorithm improved accuracy and interpretation in most cases.

For future work, we can apply different meta-heuristic algorithms. Moreover, we can use faster and stronger local search. Also, other evaluation measures can be applied to improve rule sets.

References

[1] A. Fernandez, S. Garcıa, J. Luengo, E. Bernado-Mansilla, F. Herrera, Genetics-Based Machine Learning for Rule Induction: State of the Art, Taxonomy, and Comparative Study, IEEE Transaction on Evolutionary Computation, Vol. 14, pp. 913 – 941, 2010.

[2] H. Su, Y. Yang, L. Zhao, Classification rule discovery with DE/QDE algorithm, Journal of Expert Systems with Applications, Vol. 37, pp. 1216–1222, 2010.

[3] R. Sikora: Learning controls strategies for chemical process: a distributed approach, IEEE Export, Vol.7, pp. 35–43, 1992.

[4] R. Sikora, M. Shaw, A doubled-layered learning approach to acquiring rules or classification: integrating genetic algorithms with similarity-based learning, ORSA Journal of Computing, Vol. 6, pp. 174–187, 1994.

[5] I. Lee, R. Sikora, M. Shaw, A genetic algorithm based approach to flexible flow-line scheduling with variable lot sizes, IEEE Transactions of Systems, Man, and Cybernetics, Vol. 27, pp. 36–54, 1995.

[6] R. Sikora, S. Piramuthu, An intelligent fault diagnosis system for robotic machines, International Journal of Computational Intelligence and Organizations, Vol. 1, pp. 144-153, 1996.

[7] R. R. F.Mendes, F. B. Voznika, A.A. Freitas, J. C. Nievola, Discovering fuzzy classification rules with genetic programming and coevolution, in: Proc. 5th Eur. Conf. on Principles of Data Mining and Knowledge Discovery in: Lecture Notes in Artificial Intelligence, Vol. 2168, pp. 314–325, 2001.

[8] J. Koza, Genetic Programming on the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, London. 1992.

[9] E. Bernado-Mansilla, M.J. Garella-Guiu, Accuracy-based learning classifier systems, Models, analysis and applications to classification tasks, Evolutionary Computation, Vol. 11, pp. 209–238, 2003.

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

[11] P. C. Chang, C.H. Liu, Y.W. Wang, A hybrid model by clustering and evolving fuzzy rules for sales decision support in printed circuit board industry, Decision Supports System, Vol. 42, pp. 1254–1269, 2006.

[12] C. W. M. Yuen, W. K. Wong, S.Q. Qian, L.K. Chan, E.H.K. Fung, A hybrid model using genetic algorithm and neural network for classifying garment defects, Journal on Expert Systems with Applications, Vol. 36, pp. 2037–2047, 2009.

[13] Faraoun K., M.Boukleif, Genetic programming approach for multi-category pattern classification applied to network intrusions detections, International Journal of Computational Intelligence, Vol. 6, pp 77-100, 2006.

[14] W. K. Wong, X. H. Zeng, W.M.R. Au: A decision support tool for apparel coordination through integrating the knowledge-based attribute evaluation expert system and the T–S fuzzy neural network, International Journal of Expert Systems with Applications, Vol. 36, pp. 2377-2390, 2009.

[15] B. K. Sarkar, S.S. Sana, A hybrid approach to design efficient learning classifiers, Journal of Computers and Mathematics with Applications, Vol. 58, pp. 65–73, 2009.

[16] M. Li, Z. Wang, A hybrid coevolutionary algorithm for designing fuzzy classifiers, Journal of Information Sciences, Vol. 179, pp. 1970-1983, 2009.

[17] S. Dehuri, S. Patnaik, A. Ghosh, R. Mall, Application of elitist multi-objective genetic algorithm for classification rule generation, Journal of Applied Soft Computing, Vol. 8, pp. 477–487, 2008.

[18] D. Meng, Z. Pei: Extracting linguistic rules from data sets using fuzzy logic and genetic algorithms, Journal of Neuro computing, Vol. 78, pp. 48–54, 2012.

[19] P. G. Kumar, T. A. A. Victoire, P. Renukadevi, D. Devaraj, Design of fuzzy expert system for microarray data classification using a novel Genetic Swarm Algorithm, Journal of Expert Systems with Applications, Vol. 39, pp. 1811-1821, 2012.

[20] B.K. Sarkar, S.S. Sana; A genetic algorithm-based rule extraction system, Journal of Applied Soft Computing, Vol. 12, pp. 238-254, 2012.

[21] B. K. Sarkar, S. S. Sana, K. Chaudhuri; Selecting informative rules with parallel genetic algorithm in classification problem, Applied Mathematics and Computation, Vol. 218, pp.3247-3264, 2011.

[22] J.H. Ang, K.C. Tan, A.A Mamum, An evolutionary meme tic algorithm for rule extraction, Expert Systems with Applications, Vol. 37, pp. 1302–1315, 2010.

[23] J. Roman Cano, F. Herrera, M. Lozano, Evolutionary stratified training set selection for extracting classification rules with trade off precision-interpretability, Data & Knowledge Engineering, Vol. 60, pp. 90–108, 2007.

[24] K.Y. Fung, C.K. Hwong, K.W.M Siu, K.M. Yu, A multi-objective genetic algorithm approach to rule mining for affective product design, Expert Systems with Applications, Vol. 39, pp. 7411–7419, 2012.

[25] T. Sousa, A. Silva, A. Neves, Particle swarm based data mining algorithms for classification tasks, Parallel Computing, Vol. 30, pp. 267–783, 2004.

[26] X. Zhao, J. Zeng, Y. Gao, Y. Yang, Particle swarm algorithm for classification rule generation, Sixth International conference on Intelligent System Design and Application, pp. 957-962, 2006.

[27] S. Li, C. Chen, J.W. Li, A Multi-objective Particle Swarm Optimization Algorithm for Rule Discovery, Third International conference of Intelligent Information Hiding and Multimedia Signal Processing, pp. 597-600, 2007.

[28] N. K. Khan, A. RaufBaig, M. A. Iqbal, A new discrete PSO for data classification, International Conference on Information Science and Applications (ICISA), 2010, pp. 1-6, 2010.

[29] C. Rania, S. N. Deepa: PSO with mutation for fuzzy classifier design, Procedia Computer Science, Vol. 2, pp. 307-313, 2010.

[30] N. Holden, A.A. Freitas: A hybrid PSO/ACO algorithm for discovering classification rules in data mining**,** Journal of Artificial Evolution and Applications, Vol.2008,No. 2 ,2008.

[31] C. L. Black, C. J. Mers, UCI Repository of Machine Learning Database, Department of Information and Computer Science, University of California, Irvan, 1990.

[32] B. Shabani, H. Sajedi, YABAC4.5: Yet Another Boosting Approach for C4.5 Algorithm, The 3rd International Conference on Contemporary Issues in Computer and Information Sciences, pp. 281-285, 2012.

[33] B. Lu, X. Wang, Y. Yang, H. Zhao, Learning from imbalanced data sets with a Min-Max modular support vector machine, Front, Eectr. Electron. Eng China, Vol. 6, pp. 56-71, 2011.

[34] K.C. Tan, Q. Yu, J.H. Ang, A Dual-Objective Evolutionary Algorithm for Rules Extraction in Data Mining, Computational Optimization and Applications, Vol. 34, pp.273-294, 2006.

[35] O. Dunn, Multiple comparisons among means, Journal of the American Statistical Association, Vol. 56, pp. 52–64, 1961.

[36] A. Estabrooks, T. Jo, N. Japkowicz, A multiple resampling method for learning from imbalanced data sets, Comput. Intelligence, 20, 2004, pp18–36.

[37] T. Fawcett, F.J. Provost, Adaptive fraud detection, Data Mining Knowledge Discovery, Vol. 1, pp. 291–316, 1997.

[38] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Amer. Statist. Assoc, vol. 32, pp.675–701, 1937.

[39] A. Fernandez, S. Garcia, M. Jose del Jesus, F. Herrera, A study of the behavior of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets, Fuzzy Sets and Systems, Vol. 159, pp. 2378 – 2398, 2008.

[40] G. Batista, R. Prati, M. Monard, A study of the behavior of several methods for balancing machine learning training data, SIGKDD Explorations, Vol. 6, pp. 20–29, 2004.

[41] J. Han, M. Kamber, Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 2005.

[42] R. Bansal, K. Srivastava, A memetic algorithm for the cyclic anti-band width maximization problem, Soft Compute, pp.397-412, 2011.

[43] J.H. Ang, K.C. Tan, A.A. Mamum, An evolutionary memetic algorithm for rule extraction, Expert Systems with Applications, Vol. 37, pp. 1302–1315, 2010.

[44] H. Wang, I. Moon, S. Yang, D. Wang, A memetic particle swarm optimization algorithm for multimodal optimization problems, Information Sciences, Vol. 197, pp.38–52, 2012s.