2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Iterative Mixed Integer Programming Model for Fuzzy Rule-Based Classification Systems

Shahab Derhami and Alice E. Smith

Abstract— **Fuzzy rule based systems have been successfully applied to the pattern classification problem. In this research, we proposed an iterative mixed-integer programming algorithm to generate fuzzy rules for fuzzy rule-based classification systems. The proposed model is capable of assigning the attributes to the antecedents of rules so that their inclusion enhances the accuracy and coverage of that rule. To generate several diverse rules per class, the integer programming model is run iteratively and all samples predicted correctly are temporarily removed from the training dataset in each iteration. This process ensures that subsequent rule covers new samples in the associated class. The proposed model was evaluated on the benchmark datasets from the UCI repository and this comparative study verifies that this approach extracts accurate rules and has advantage over conventional approaches for high dimensional datasets.**

I. Introduction

The capability of Fuzzy Systems in establishing a precise model has been confirmed, especially when the system deals with ambiguous and imprecise data. Recently, Fuzzy Rule-based Systems (FRBSs) have been successfully applied to the pattern classification problem [1], [2]. Their potential power to encompass nonlinear and complex relations in a system by using linguistic and easily understood terms makes them appropriate tools to extract embedded data in classification problems. The main advantage of FRBSs is that their interpretable linguistic models are easily understood by the users and can be obtained either from an expert, analyzing data by mathematical approaches, or both.

Different approaches have been proposed for designing optimal fuzzy classifiers such as artificial neural networks [3]-[5], heuristic approaches [2], clustering methods [6], [7] and genetic algorithms (GA) [8]-[14]. Among all employed methods, GA has been broadly used for Fuzzy Rule Based Classification Systems (FRBCSs), recently. It has been applied in FRBCSs in two different approaches.

In the first approach, GA is employed to generate fuzzy rules that have common fuzzy sets in share. In this method, number of fuzzy sets per attribute and their membership functions are determined in advance and GA is applied to find the best set of the rules based on the predefined fuzzy sets [8], [12], [15]-[17]. The main disadvantage of this approach is that it may not obtain good solutions because fuzzy sets and rules have mutual relationships and optimizing one without considering the other one may lead to poor outcomes. To overcome this issue, we will solve the model for different fuzzy sets and then select the design that has the most accurate outcome.

In the second approach, GA is applied to generate fuzzy rules that have their own definition of membership functions. In this approach both fuzzy rules and membership functions are optimized simultaneously. The major disadvantage of this approach is that it has a larger decision space and finding an optimal solution is more difficult [13], [18], [19].

Aydogan *et al*, proposed a hybrid GA for FRBCSs that follows the first mentioned approach. They applied a GA to obtain optimal fuzzy rules over predefined fuzzy sets and then among the rule pool, the set of most accurate rules that covers almost whole training dataset is selected by an integer programming model.

Li and Wang, proposed a hybrid GA that pursues the second mentioned approach [13]. Their proposed model optimizes fuzzy rules and membership functions simultaneously. They developed a local search algorithm that improves the quality of the rules obtained by a GA. Comparative study of their proposed hybrid method shows that their algorithm performs well on the well known benchmark datasets.

In this study we propose an iterative Mixed-Integer Programming Model (MIPM) to obtain optimal fuzzy rules for FRBCSs. The model obtains optimal fuzzy rules for a predefined set of fuzzy sets and membership functions. The main advantage of MIPM is that the antecedents of the optimal rule consist of only the attributes that their involvement in the optimal rule contributes in enhancing the accuracy and coverage of it. It means the antecedent part of the optimal fuzzy rule is a main decision variable in our model and the goal is to determine which attributes and fuzzy sets should be included in the antecedent part of the optimal rule. To incorporate the degree of accuracy and coverage of a rule in the MIPM, we develop a new linear approximation for these terms that can be used in integer programming model.

The remainder of this paper is organized as follows. Some basic concepts of FRBCSs are introduced in section II. The MIPM notation and formulation are described in section III. Computational results and comparative results are reported in section IV and finally a summary of this research and the direction of the future works follow in section V.

II. Fuzzy Rule-Based Classifiers

Suppose that the training dataset contains m training patterns or samples with n attributes. It means every sample of this dataset can be presented as an n-tuple $S_i=(s_{i1}, \ldots, s_{in})$, i

Shahab Derhami is PhD student at Industrial and Systems Engineering Department at Auburn University, AL, USA.
(E-mail: Shderhami@Auburn.edu).
Alice E. Smith is W. Allen and Martha Reed Professor of the Industrial and Systems Engineering Department at Auburn University, AL, USA. (Email: Smithae@auburn.edu).

= 1,2,. . .,m, where s_{ij} is the value of the jth attribute ($j = 1,2,. . .,n$) of the ith training sample. A fuzzy rule considered in this research is in the form of the following rule:

$$Rule\ R_q: if\ s_1\ is\ A_{1k^1}\ and\ \dots and\ s_n\ is\ A_{nk^n}\ then\ class\ C$$

where $S=(s_1, ..., s_n)$ is the input vector of an example, $A_{jk}j$ is the kth fuzzy linguistic variable of jth fuzzy antecedent of rule R_q. Figure 1 represents a fuzzy set consisting of five fuzzy linguistic variables which have normal symmetric triangular membership functions. In this research we use similar fuzzy sets with two to six fuzzy linguistic variables. To evaluate the degree of association of a pattern to a rule, the compatibility degree of that pattern to all antecedents of the rule should be computed. Suppose that $\mu_{A_{jk}j}(s_{ij})$ is the membership value of the jth attribute of the input sample i in the kth linguistic variable of this attribute, then the degree of compatibility of sample S_i with the fuzzy rule R_q is obtained as:

$$\mu_{A_q}(S_i) = \mu_{A_{1k^1}}(s_{i1}).\mu_{A_{2k^2}}(s_{i2})\dots\mu_{A_{nk^n}}(s_{in}),\ \ i = 1\dots m \quad (1)$$

For a normal fuzzy set, the degree of compatibility of a sample to a rule is between zero and one in which zero shows the sample does not belong to the rule and one shows the rule covers the pattern perfectly, *i.e.* the sample has membership values of one in all antecedents of the rule. Different factors have been proposed to evaluate effectiveness of a rule in FRBCSs. Among these factors, the two that have most frequently been used by researchers to evaluate accuracy and coverage of a rule are denoted by confidence and completeness [13].

Confidence measures accuracy of a rule and is defined as:

$$conf\,(R_q) = \frac{\eta^+(R_q)}{\eta^+(R_q) + \eta^-(R_q)} \quad (2)$$

where $\eta^+(R_q)$ and $\eta^-(R_q)$ are the sums of the degree of compatibility of the samples that are predicted correctly and incorrectly with the rule R_q, respectively, and are computed as:

$$\eta^+(R_q) = \sum_{X_i \in C_q} \mu_{A_q}(S_i) \quad (3)$$

$$\eta^-(R_q) = \sum_{X_i \notin C_q} \mu_{A_q}(S_i) \quad (4)$$

Here C_q represents the class that rule R_q belongs to. Completeness computes the proportion of the samples that are correctly covered by the rule R_q, and is defined as:

$$comp\,(R_q) = \frac{\eta^+(R_q)}{N_{C_q}} \quad (5)$$

where N_{C_q} is number of samples of training dataset that belong to class C_q. These two factors are used to control the accuracy and coverage of the optimal rules in MIPM. Using the compatibility degree of a rule as demonstrated in expression (5) makes our model nonlinear because although the membership values of samples to all fuzzy sets of the attributes are known, the compatibility degree of a pattern with the optimal rule depends on the set of all active antecedents of that rule, which is calculated by the following nonlinear term:

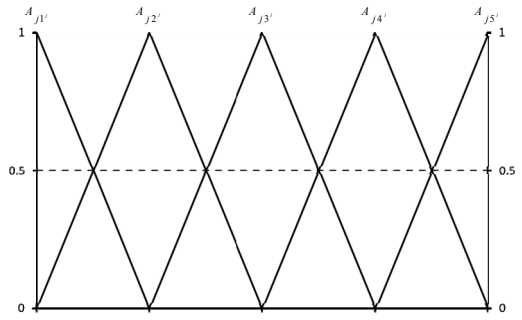

Fig. 1. Fuzzy membership functions of attribute j with five fuzzy linguistic labels.

$$\mu_{A_q}(S_i) = \prod_{j\in J,\, k\in K^j} \mu_{A_{jk}}(s_{ij})\, x_{jk}\ \ j = 1\dots n,\ k = 1\dots k^j \quad (6)$$

where x_{jk} is a binary decision variable and equals 1 if antecedent k of attribute j is selected in the optimal rule and 0 otherwise. To prevent using this nonlinear term in our model, we approximate the compatibility degree of a pattern i with a rule R_q by the following equation:

$$\mu'_{A_q}(S_i) = \begin{cases} \dfrac{\mu_{A_{1k^1}}(s_{i1}) + \dots + \mu_{A_{nk^n}}(s_{in})}{\eta(R_q)} & \mu_{A_{jk^j}}(s_{ij}) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where $\eta(R_q)$ is the number of antecedents of rule R_q. Like $\mu_{A_q}(S_i)$, $\acute{\mu}_{A_q}(S_i)$ would have a value between 0 and 1 in which 0 shows the rule does not cover the sample and values close to 1 show the sample is more compatible with the rule. Using this approximation, the compatibility degree in MIPM would look like the following equation:

$$\mu'_{A_q}(S_i) = \begin{cases} \dfrac{\sum_{j\in J}\sum_{k\in K^j} \mu_{A_{jk}}(S_{ij})\, x_{jk}}{\eta(R_q)} & \mu_{A_{jk^j}}(S_{ij}) > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

This equation is still not linear because the denominator is still a decision variable in our model. This is due to the fact that the model is capable of choosing the best set of antecedents for the optimal rule and hence $\eta(R_q)$ is a decision variable that depends on the number of antecedents of a rule; however, this term is used as a constraint in our model to force the optimal rule to satisfy minimum levels of confidence and completeness. Therefore, it is simply converted to a linear constraint as its right hand side is a parameter.

III. MODEL

In this section we describe the settings of the model and provide the mathematical formulations of the MIPM.

A. Notation

The parameters and sets that used in the MIPM are listed in the Table I. The following are the decision variables used in the model:

TABLE I
NOTATION OF PARAMETERS

I	Set of samples
J	Set of attributes
K	Set of fuzzy sets for each attribute
C	Set of classes
μ_{ijk}	Membership value of sample i for kth fuzzy antecedent of attribute j
N_c	Number of Samples in class c
B_{ic}	Equal to 1 if sample i belongs to class c and 0 otherwise
α	Minimum level of confidence
σ	Minimum level of completeness
ε	Arbitrary small enough number
$M_1, \ldots, M_5$	Arbitrary large enough number

$$x_{jk} = \begin{cases} 1 & \text{if fuzzy linguistic variable } k \text{ of attribute } j \text{ is selected as the } j\text{th antecedent of the optimal rule,} \\ 0 & \text{otherwise,} \end{cases}$$

$$CL_c = \begin{cases} 1 & \text{if the optimal rule belongs to class } c, \\ 0 & \text{otherwise,} \end{cases}$$

$$\omega_{ic} = \begin{cases} 1 & \text{if sample } i \text{ belongs to the optimal rule of class } c, \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{ic} = \begin{cases} 1 & \text{if sample } i \text{ does not belong to the optimal rule,} \\ 0 & \text{otherwise,} \end{cases}$$

γ_i : degree of compatibility of sample i with the optimal rule without considering classes, i.e., compatibility degree of corrected and uncorrected predictions with the optimal rule.

η_{ic} : degree of compatibility of sample i with the optimal rule that belongs to class c, i.e., compatibility degree of corrected predictions with the optimal rule.

B. Formulation:

$$Max \quad \sum_{i\in I}\sum_{c\in C} \omega_{ic} \tag{9}$$

s.t.

$$\sum_{k\in K} x_{jk} \le 1 \qquad \forall j \in J \tag{10}$$

$$\sum_{c\in C} CL_c \le 1 \qquad \forall j \in J \tag{11}$$

$$\sum_{j\in J}\sum_{k\in K} \mu_{ijk}\, x_{jk} + M_1 y_i \ge 0 \qquad \forall i \in I \tag{12}$$

$$\gamma_i \le M_2(1 - y_i) \qquad \forall i \in I \tag{13}$$

$$\sum_{j\in J}\sum_{k\in K} \mu_{ijk}\, x_{jk} - M_3(1 - y_i) \le -\varepsilon \qquad \forall i \in I \tag{14}$$

$$\gamma_i \le \sum_{j\in J}\sum_{k\in K} \mu_{ijk}\, x_{jk} + M_4 y_i \qquad \forall i \in I \tag{15}$$

$$\gamma_i \ge \sum_{j\in J}\sum_{k\in K} \mu_{ijk}\, x_{jk} \qquad \forall i \in I \tag{16}$$

$$\eta_{ic} \le \gamma_i \qquad \forall i \in I,\ \forall c \in C \tag{17}$$

$$\eta_{ic} \le M_5 \beta_{ic} CL_c \qquad \forall i \in I,\ \forall c \in C \tag{18}$$

$$\omega_{ic} \le \eta_{ic} \qquad \forall i \in I,\ \forall c \in C \tag{19}$$

$$\sum_{i\in I}\sum_{c\in C} \eta_{ic} \ge \alpha \sum_{i\in I} \gamma_i \tag{20}$$

$$\sum_{i\in I}\sum_{c\in C} \frac{\eta_{ic}}{N_c} \ge \sigma \sum_{j\in J}\sum_{k\in K} x_{jk} \tag{21}$$

In this model, objective (9) is to find a rule with the maximum number of corrected predictions. Constraint (10) ensures that at most one fuzzy linguistic variable is selected per attribute. Constraint (11) guarantees that the optimal rule belongs to only one class. Constraints (12), (13), (14), (15) and (16) determine the compatibility of the patterns with the optimal rule for both corrected and uncorrected predictions. As demonstrated in (8), even if one attribute of a sample does not belong to the fuzzy linguistic variable of the associated antecedent in the optimal rule then the compatibility degree of that pattern with the rule would be zero. Also, these sets of constraints ensure that just those attributes included as antecedents in the optimal rule improves the accuracy and coverage of the rule. In the other words, this set of constraints decide which fuzzy linguistic variable of which attribute should be included as an antecedent in the optimal rule to maximize the coverage and accuracy of the rule.

Constraints (17) and (18) determine the compatibility degree of the samples that are predicted. Constraint (19) and the objective function together maximize the number of samples that covered by the rule. Constraints (20) and (21) restrict the optimal rule to meet the minimum acceptable levels of completeness and confidence.

This mixed integer programming model obtains a rule which covers the largest proportion of the samples belonging to its class and meets the minimum level of accuracy and completeness defined by (2), (5) and (8). Usually one rule cannot cover all samples belonging to a particular class especially when high dimensional classification problems are considered. To accurately classify all samples belonging to a particular class, we need to extract all of the embedded knowledge in the training dataset by finding as many as rules as needed to cover almost all samples belonging to that class. To generate multiple rules for each class we use MIPM as the main part of an iterative algorithm that generates a new optimal rule per iteration and removes all correctly covered samples from the dataset to prepare it for extracting the next rule in the next iteration. This procedure is as follow:

After initializing parameters of the MIPM, it is run for the first class. To force the model to obtain a rule belonging to the first class, the following constraint is added to MIPM:

$$CL_c = 1 \tag{22}$$

where $c=1$ at this point. If a nonzero solution exists then that rule is added to the rule pool and all samples that have been covered correctly by this rule are temporarily removed from the training dataset. Then MIPM is run again to obtain the next rule belonging to this class. If in any iteration a nonzero solution does not exist then the accuracy level (α) is diminished by 5 percent and the algorithm continues. This process continues until 95 percent of the samples belonging to the first class being removed from the dataset. It means this process repeats to cover at least 95 percent of the samples belonging to a particular class. Then all the parameters including the accuracy level are reinitialized and the algorithm is restarted by selecting the next class. This process is repeated for all classes in the training dataset. Table II represents the overall pseudo-code of the proposed algorithm.

TABLE II
THE OVERALL PSEUDO-CODE OF THE PROPOSED ALGORITHM

```
Initialize parameters
Design fuzzy sets
Calculate membership values for all samples
For all c belonging to classes in the training dataset do
    Set α to its maximum desired value
    Add the following constraint to MIPM: CL_c=1
    While (remaining number of samples belonging to c is bigger than
    5% of initial number of samples belonging to c) do
        Run MIPM
        If an optimal solution exists Then
            Add the new solution to the rules pool
            Remove all samples correctly covered by the rule from the
            training set.
        Else
            Reduce α
    Continue
    Reset the training dataset by restoring all removed samples
    Remove the added constraint
Continue
Prune the rule pool
```

TABLE III
DATASET DESCRIPTIONS

Dataset	Number of attributes	classes	Patterns
Glass	9	6	214
HillValey1	100	2	1212
HillValey2	100	2	1212
Iris	4	3	150
Libras Mov.	90	15	360
Sonar	60	2	208
Wdbc	30	2	569
Wine	13	3	178

TABLE IV
PARAMETER SETTINGS

Parameter	Value
α	20%-100%
σ	20%-80%
K	2-6
$M_1,\ldots,M_5$	1000
ε	0.0001

C. Rule reduction strategy

Constraint (21) ensures that the optimal rule meets the minimum level of completeness (σ). Determining an appropriate value for σ generally depends on the training dataset and can be done by trial and error. Higher values of σ (close to one) restrict the model to obtain a rule that covers most samples belonging to its associated class. Our experiments show that this may become an issue when MIPM attempts to obtain the second or subsequent rules in the same class as they may cover much fewer samples and have smaller completeness in compared with the first generated rule. Therefore a large σ may lead to infeasible solutions for second or subsequent rules. On the other hand, assigning small values (close to zero) to σ may make the constraint (21) so loose that the last generated rules in the same class only cover a few samples or even in some extreme cases just one sample.

One approach to overcome this challenge is to find the most appropriate value of σ for each dataset by trial and error. The disadvantages of this approach are that extensive computational effort is required and for every dataset a different value of σ would be used.

The other approach is to assign a relatively small value to σ in order to avoid infeasibility and then at the final step, prune inaccurate rules that cover a small portion of the dataset by removing them from the rule pool. We applied this latter approach in this study and developed a simple algorithm that removes the rules that cover just one sample in the training dataset or have an accuracy of less than 20 percent.

D. Fuzzy Reasoning Method

The fuzzy reasoning system determines which rule most accurately classifies a particular sample. In this study the maximum matching method was used as a fuzzy inference method. This method classifies a sample by using a rule that has highest compatibility degree with that sample and ignores the information that is given by the other rules. Equation (7) is used to calculate the compatibility degree of a sample with the rules.

There are usually a few samples in the training or test datasets that cannot be classified by any rule. It means the compatibility degrees of these samples with all of the rules are zero and they are termed unclassified samples. We use the following strategy to prevent having unclassified samples in this research. If the compatibility degree of a sample with all rules computing by (7) yields zero, then (7) is relaxed to the following equation to compute the compatibility degree for such samples.

$$\mu''_{A_q}(S_i) = \frac{\mu_{A_{1k^1}}(s_{i1}) + \cdots + \mu_{A_{nk^n}}(s_{in})}{\eta(R_q)} \tag{22}$$

IV. EXPERIMENTAL RESULTS

A. Datasets

To validate and evaluate the effectiveness of the proposed model, we examined it on the well-known datasets obtained from UCI (University of California at Irvine) Machine Learning Repository [20]. We examined our model on eight datasets. Among these datasets five of them include 30 to 100 attributes. We selected these high dimensional datasets to examine the performance of the proposed model on complex problems that have numerous attributes. The important characteristics of these datasets are presented in Table III.

B. Experimental setup

A 10 fold cross validation procedure was used for all experiments. Since the proposed model is a deterministic model, this procedure was performed three times by using three different seeds to obtain 30 independent datasets. Therefore the proposed model was run 30 times on each dataset and the averages of the results are reported.

The parameters involved in the MIPM are presented in Table IV. MIPM was coded in OPL language using IBM ILOG CPLEX Optimization Studio 12.5 and ran in an Intel Core i7 CPU (3.4GHz) desktop computer.

TABLE V
PREDICTIVE ACCURACY FOR DIFFERENT DESIGNS OF FUZZY SETS

Dataset	2 Fuzzy Sets			3 Fuzzy Sets			4 Fuzzy Sets			5 Fuzzy Sets			6 Fuzzy Sets			Comp (min)
	%Tra	%Test	#Rules	%Tra	%Test	#Rules	%Tra	%Test	#Rules	%Tra	%Test	#Rules	%Tra	%Test	#Rules	
Glass	49.42	49.52	4.10	62.10	57.14	11.23	75.47	62.70	20.63	81.01	**66.98**	25.30	73.97	57.86	24.05	4
HillValey1	50.38	49.59	2.00	47.77	50.00	1.87	51.21	51.32	2.50	52.64	51.35	4.00	52.13	**51.84**	4.03	63
HillValey2	50.77	49.86	2.00	50.93	50.83	2.00	52.38	**51.82**	3.30	51.93	51.10	4.00	51.63	50.99	4.00	64
Iris	84.94	84.89	3.00	86.99	85.56	5.47	97.23	95.56	6.03	97.04	**96.44**	8.87	95.41	93.33	7.07	3
Libras Mov.	85.30	23.33	15.00	94.38	58.61	47.20	97.33	69.91	37.33	97.45	68.70	38.87	97.42	**70.19**	36.60	52
Sonar	67.78	52.75	2.05	96.30	76.50	10.77	97.76	76.17	9.93	97.72	**79.83**	9.67	97.89	73.17	10.63	27
Wdbc	68.46	68.01	2.00	94.71	94.14	4.80	96.04	**94.73**	6.03	96.03	94.26	6.40	96.38	93.85	7.63	8
Wine	86.33	86.52	3.00	95.13	88.20	11.97	96.53	94.19	5.93	98.15	**94.76**	6.80	98.02	92.32	7.63	3
Average	67.92	58.06	4.14	78.54	70.12	11.91	82.99	74.55	11.46	84.00	**75.43**	12.99	82.86	72.94	12.71	28.00

C. Analysis of designing fuzzy sets

The initial part of performing experiments is to design fuzzy sets and compute the membership values for all samples of the training dataset. Designing fuzzy sets is not a part of the optimization in MIPM and fuzzy membership functions are instead given to the model as input parameters. Therefore any type of membership functions is allowed to be defined. In this study, we assumed all fuzzy sets have normal symmetric triangular membership functions like Figure 1. In order to find the proper number of fuzzy sets that maximizes predictive accuracy, we carried out all experiments for two to six fuzzy sets per attribute. Percentages of the predictive accuracy for the training datasets (%Tra), test datasets (%Test), number of generated rules (#Rules) and computational times (Comp) in minutes are presented in Table V. The experimental results show that the predictive accuracy and number of generated rules enhance as the number of fuzzy sets increases. This behavior was expected because the fuzzy sets cover smaller intervals of their associated attributes when the number of fuzzy sets increases. That is, each fuzzy set covers a smaller portion of the solution space and, as a consequence, more rules are required to cover the dataset. On the other hand, the common interval between two adjacent fuzzy sets shrinks as the number of fuzzy sets increases. This means the fuzzy sets become less ambiguous because outlier data is removed as a result of this shrinkage of the membership functions. This accuracy in the fuzzy sets leads to an increase in predictive accuracy of the fuzzy rules. Table V presents that the highest average predictive accuracy was achieved by increasing the number of fuzzy sets to five. The average predictive accuracy reduces as number of fuzzy sets increases to six. This reduction occurs because by dividing an attribute into too many fuzzy sets, each fuzzy set covers a small interval of its associated attributes. As these intervals become smaller, the fuzzy sets converge to crisp sets. In such a case, FRBCS loses its capability to accurately classify the problem. Our experiment on the aforementioned datasets shows that MIPM obtains the highest predictive accuracy with five fuzzy sets.

In terms of computational effort, Table V shows that the average computational time of MIMP increases as the number of attributes increases in the datasets. We expected this behavior because high dimensional problems demand more computational resources.

D. Comparative analysis with other FRBCS methods

In order to demonstrate the competitive performance of the proposed algorithm, we compare the predictive accuracy of the proposed algorithm on the training and test datasets as well as the number of generated rules with the ones obtained by three different genetic algorithms: 2SLAVE [21], FRBCS-GP [22] and GP-COACH [22]. Table VI compares the predictive accuracy and number of rules obtained by MIPM and the above mentioned classifiers on the training and test datasets. The results of the comparative algorithms were obtained from [22]. As it can be seen from the table, MIPM achieved the highest predictive accuracy in the Glass, Libras Mov. and Sonar datasets. In the other datasets, the results of MIPM are very close to the highest predictive accuracy obtained by the other methods. Table VI shows that MIPM achieved higher average predictive accuracy than the

TABLE VI
PREDICTIVE ACCURACY OF MIPM AND OTHER FRBCS METHODS

Dataset	2SLAVE			FRBCS-GP			GP-COACH			MIPM		
	%Tra	%Test	#Rules	%Tra	%Test	#Rules	%Tra	%Test	#Rules	%Tra	%Test	#Rules
Glass	49.29	44.39	8.80	61.28	56.61	23.47	71.26	65.33	17.43	81.01	**66.98**	25.30
HillValey1	52.52	51.76	6.63	50.48	49.78	26.93	53.96	**52.89**	7.27	52.13	51.84	4.03
HillValey2	52.53	51.21	6.40	51.28	50.69	34.33	55.68	**53.99**	6.90	52.38	51.82	3.30
Iris	94.67	94.67	3.93	97.65	97.11	3.00	97.78	**97.56**	3.23	97.04	96.44	8.87
Libras Mov.	33.15	25.83	25.53	56.24	47.69	49.77	74.22	45.56	113.93	97.42	**70.19**	36.60
Sonar	78.45	70.72	9.33	83.30	71.15	20.97	80.25	67.48	14.03	97.72	**79.83**	9.67
Wdbc	92.42	91.80	5.47	95.60	**95.02**	16.30	95.09	93.90	4.90	96.04	94.73	6.03
Wine	92.22	91.53	5.73	95.84	91.13	9.60	98.96	**95.10**	7.57	98.15	94.76	6.80
Average	68.16	65.24	8.98	73.96	69.90	23.05	78.40	71.48	21.91	83.99	**75.82**	12.58

other methods. In terms of the number of rules, MIPM generates fewer rules, on average, than FRBCS-GP and GP-COACH but produces more rules than 2SLAVE. Comparing the predictive accuracy of MIPM and 2SLAVE shows that in all datasets MIPM achieved more accurate results than 2SLAVE. It shows that the higher number of rules produced by MIPM did not lead to over fitting. In fact, the iterative structure of MIPM enables it to produce more accurate rules since it uses an exact optimization technique that makes it capable of recognizing patterns in the datasets. The fact that our model achieved the highest predictive accuracy on the Libras Mov. and Sonar datasets and obtained very close results to the most accurate model on the HillValey1 and HillValey2 datasets (which are among the most high dimensional datasets in the literature) provides evidence that our model is very capable of extracting accurate rules from complicated and high dimensional datasets.

V. Conclusions

In this paper we developed an iterative mixed-integer programming model to generate fuzzy rules for fuzzy rule-based classification problems. The proposed model is capable of assigning attributes to the antecedents of rules in an optimal manner so that their inclusion maximally improves the accuracy and coverage of that rule. To linearize the commonly used nonlinear expression to compute accuracy and coverage of rules, we developed a linear approximation to estimate accuracy and coverage of a rule. The iterative algorithm generates an optimal rule per iteration and removes correctly predicted samples from the training dataset in that iteration. This process guarantees that repetitive solution is not generated and new useful knowledge is discovered during each iteration.

Comparative study of the proposed algorithm on benchmark datasets from UCI repository verifies that it extracts accurate rules from training datasets and results in high predictive accuracy. Although, the proposed algorithm performed well on the training datasets, there are still possibilities to improve its effectiveness in future studies. Developing a multi objective model and a branch and cut technique to solve large problems are two directions to improve this approach.

VI. Acknowledgment

The authors are very grateful to the anonymous reviewers for their helpful comments.

References

[1] L. Hu, H.D. Cheng, M. Zhang, "A high performance edge detector based on fuzzy inference rules", Information Sciences, vol. 177(21), pp. 4768–4784, 2007.

[2] H. Ishibuchi, K. Nozaki, H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification", Fuzzy Sets and Systems, vol. 52(1), pp. 21–32, 1992.

[3] D. Chakraborty, N.R. Pal, "A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification", IEEE Transactions on Neural Networks, vol. 15(1), pp. 110–123, 2004.

[4] C.T. Lin, C.S.G. Lee, "Neural network-based fuzzy logic control and decision system", IEEE Transactions on Computers, vol. 40(12), pp. 1320–1336, 1991.

[5] D. Nauck, R. Kruse,"A neuro-fuzzy method to learn fuzzy classification rules from data", Fuzzy Sets and Systems, vol. 89(3), pp. 277–288, 1997.

[6] S. Abe, R. Thawonmas, "A fuzzy classifier with ellipsoidal regions", IEEE Transactions on Fuzzy Systems, vol. 5(3), pp. 358–368, 1997.

[7] C.-Y. Lee, C.-J. Lin, H.-J. Chen, "A self-constructing fuzzy CMAC model and its applications", Information Sciences, vol. 177(1), pp. 264–280, 2007.

[8] E. K. Aydogan, I. Karaoglan, P. M. Pardalos, "hGA: Hybrid genetic algorithm in fuzzy rule-based classification systems for high-dimensional problems", Applied Soft Computing, vol. 12, pp. 800–806, 2012.

[9] R. Alcal, J. Alcala-Fdez, F. Herrera, J. Otero, "Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation", International Journal of Approximate Reasoning, vol. 44(1), pp. 45–64, 2007.

[10] A. F. Gómez-Skarmeta, M. Valdés, F. Jiménez, and J. G. Marín-Blázquez, "Approximative fuzzy rules approaches for classification with hybrid-GA techniques", Information Sciences, vol. 136(1–4), pp. 193–214, 2001.

[11] Y.-C. Hu, "Finding useful fuzzy concepts for pattern classification using genetic algorithm", Information Sciences, vol. 175(1–2), pp. 1–19, 2005.

[12] H. Ishibuchi, T. Yamamoto, T. Nakashima, "Hybridization of fuzzy GBML approaches for pattern classification problems", IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 35(2), pp. 359–365, 2005.

[13] M. Li, Z. Wang, "A hybrid coevolutionary algorithm for designing fuzzy classifiers", Information Sciences, vol. 179, pp. 1970–1983, 2009.

[14] E. Zhou, A. Khotanzad, "Fuzzy classifier design using genetic algorithms", Pattern Recognition, vol. 40(12), pp. 3401–3414, 2007.

[15] A. Fernández, M.J. del Jesus, F. Herrera, "Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets", International Journal of Approximate Reasoning, vol. 50, pp. 561–577, 2009.

[16] H. Ishibuchi, Y. Nojima, "Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning", International Journal of Approximate Reasoning, vol. 44(1), pp. 4–31, 2007.

[17] E.G. Mansoori, M.J. Zolghadri, S.D. Katebi, "SGERD: a steady-state genetic algorithm for extracting fuzzy classification rules from data", IEEE Transactions on Fuzzy Systems, vol. 16(4), pp. 1061–1071, 2008.

[18] M.J. del Jesus, F. Hoffmann, L. J. Navascues, L. Sanchez, "Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms", IEEE Transactions on Fuzzy Systems, vol. 12(3), pp. 296–308, 2004.

[19] F. Hoffmann, B. Baesens, C. Mues, T. Van Gestel, J. Vanthienen, "Inferring descriptive and approximate fuzzy rules for credit scoring using evolutionary algorithms", European Journal of Operational Research, vol. 177(1), pp. 540–555, 2007.

[20] Bache, K., M. Lichman, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.

[21] A. González, R. Pérez, "Selection of relevant features in a fuzzy genetic learning algorithm", IEEE Transactions on Systems, Man and Cybernetics- PART B: CYBERNETICS, vol. 31(3), pp. 417–425, 2001.

[22] F.J. Berlanga, A.J. Rivera, M.J. del Jesus, F. Herrera, "GP-COACH: Genetic Programming-based learning of COmpact and ACcurate fuzzy rule-based classification systems for High-dimensional Problems", Information Sciences, vol. 180, pp. 1183–1200, 2010.