Applied Soft Computing 41 (2016) 91–93

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Technical Note

A technical note on the paper "hGA: Hybrid genetic algorithm in fuzzy rule-based classification systems for high-dimensional problems"

Shahab Derhami, Alice E. Smith*

Department of Industrial and Systems Engineering, Auburn University, AL, USA

ARTICLE INFO

Article history:
Received 22 May 2015
Received in revised form 13 August 2015
Accepted 1 October 2015
Available online 29 December 2015

Keywords:
Fuzzy rule-based classification systems
Integer programming
Genetic algorithms
Genetic fuzzy systems
Classification

ABSTRACT

This paper provides a corrected formulation to the mixed integer programming model proposed by Aydogan et al. (2012) [1]. They proposed a genetic algorithm to learn fuzzy rules for a fuzzy rule-based classification system and developed a Mixed Integer Programming model (MIP) to prune the generated rules by selecting the best set of rules to maximize predictive accuracy. However, their proposed MIP formulation contains errors, which are described in this technical note. We develop corrections and improvements to the original formulation and test it with non-parametric statistical tests on the same data sets used to evaluate the original model. The statistical analysis shows that the results of the correction formulation are significantly different from the original model.

1. Introduction

Aydogan et al. [1] developed a hybrid algorithm to learn fuzzy rules for Fuzzy Rule-Based Classification Systems (FRBCSs). They proposed a Genetic Algorithm (GA) that learns fuzzy rules from a data set and keeps them in a rule base. Their GA learns as many fuzzy rules as the population size for each class label. The population size was set equal to 50 in their experimental study making the size of the rule base equal to the number of class labels multiplied by 50. Once the GA terminates, the rule base is pruned by a Mixed Integer Programming model (MIP) that selects the best set of rules with the objective to maximize the total number of samples predicted correctly by the FRBCS. Their MIP model classifies samples using the normalized sum Fuzzy Reasoning Method (FRM). In the normalized sum FRM, matching degrees between a sample and all rules with the same class labels are aggregated and normalized. This normalized sum is calculated for all class labels and the class label that obtains the highest normalized sum is selected to classify the sample. The MIP model uses the same concept to classify the sample, but instead of normalizing the sum, it calculates a weighted average where the accuracy of the rules is used as their weights. This change was made to prevent having a non-linear constraint in the MIP model.

In this note we highlight the errors in the formulation of the original MIP model developed by Aydogan et al. [1] and propose a corrected formulation. For the sake of convenience, the original formulation proposed by Aydogan et al. [1] is mentioned in the following. First the definition of the sets, parameters, and decision variables are described.

C	set of classes
V	set of variables
E	set of samples
R	set of rules
α_{er}	matching degree between sample e and fuzzy rule r ($e \in E, r \in R$)
β_r	accuracy of the rule r, ($r \in R$)
C_e	class label of sample e, ($e \in E$)
C_r	class label of the rule r, ($r \in R$)
M	arbitrary large enough number
x_r	binary decision variable, $x_r = 1$ if rule r is selected ($r \in R$)
y_{ec}	binary decision variable, $y_{ec} = 1$ if sample e classified to the class label c ($e \in E, c \in C$)
Acc_e	continuous decision variable, maximum matching degree for the sample e with respect to the selected rules, ($e \in E$)

$$\text{Maximize} \quad \sum_{e \in E, c \in C: C_e = c} y_{ec} \tag{1}$$

Subject to

$$\sum_{r \in R: C_r = c} x_r \geq 1 \quad \forall c \in C \tag{2}$$

$$\sum_{c \in C} y_{ec} \leq 1 \quad \forall e \in E \tag{3}$$

$$Acc_e \geq \sum_{r \in R: C_r = c} \alpha_{er} \beta_r x_r \quad \forall e \in E, c \in C \tag{4}$$

$$y_{ec} \leq 1 - \frac{1}{M} \left(Acc_e - \sum_{r \in R: C_r = c} \alpha_{er} \beta_r x_r \right) \quad \forall e \in E, c \in C \tag{5}$$

* Corresponding author. Tel.: +1 334 844 1460; fax: +1 334 844 1381.
E-mail address: smithae@auburn.edu (A.E. Smith).

http://dx.doi.org/10.1016/j.asoc.2015.10.016
1568-4946/

$$y_{ec} \le \sum_{r \in R: C_r = c} x_r \quad \forall e \in E, c \in C \tag{6}$$

$$y_{ec} \in \{0, 1\} \quad \forall e \in E, c \in C \tag{7}$$

$$x_r \in \{0, 1\} \quad \forall r \in R \tag{8}$$

$$Acc_e \ge 0 \quad \forall e \in E \tag{9}$$

The objective function (1) aims to maximize total number of samples predicted correctly. Constraint (2) ensures that at least one rule per class label is selected. Constraint (3) guarantees that each sample is classified to not more than one class label. Constraint (4) calculates the highest matching degree between the selected rules and the samples. Constraint (5) determines the class label of the samples with enforcement of the objective function as follows. For a sample, $\left(Acc_e - \sum_{r \in R: C_r = c} \alpha_{er} \beta_r x_r\right)$ is equal to zero for the class label whose rules obtain the highest weighted sum of the matching degrees and the objective function (1) enforces y_{ec} to become one for that sample and class label. For the remaining class labels, $\left(Acc_e - \sum_{r \in R: C_r = c} \alpha_{er} \beta_r x_r\right)$ is a positive value which becomes less than one after devision by M. Therefore, the y_{ec}'s are forced to be equal to zero for those class labels. Constraint (6) ensures that a sample is not classified to a class label unless at least one of the rules belonging to that class label is selected. Constraints (7) to (9) describe the decision variable definitions.

2. Description of the errors and the correction formulation

The original MIP model proposed by Aydogan et al. [1] classifies a sample to the class label whose rules obtain the highest weighted sum of the matching degrees. However, it does not correctly classify a sample whose matching degrees with all selected rules are zero. In such a case, the right-hand side of the constraint (5) is equal to zero and Acc_e is greater than or equal to zero. As a result, $\left(Acc_e - \sum_{r \in R: C_r = c} \alpha_{er} \beta_r x_r\right)$ in constraint (6) is equal to zero for at least one class label and therefore, y_{ec} is equal to one for that class label. In other words, the original MIP model classifies samples that are not compatible with any of the selected rules to random class labels and considers these classifications as correct ones. However, obviously, those samples are properly considered unclassified samples or incorrect classifications. To prevent this error, which affects the accuracy and reliability of the MIP model, we propose to add the following constraint to the model:

$$y_{ec} \le M_2 \sum_{r \in R: C_r = c} \alpha_{er} \beta_r x_r \quad \forall e \in E, c \in C \tag{10}$$

where M_2 is an arbitrary large enough number. Constraint (10) enforces y_{ec} to be equal to zero when the weighted sum of the matching degrees between a sample and all selected rules are equal to zero. The quantity of M_2 depends on the acceptable level of accuracy and tolerance for the matching degree when it converges to zero. It determines at what level a matching degree is considered zero. Smaller values of M_2 result in a large weighted sum to be zero while the reverse is true for larger values of M_2. We suggest to set it equal to 10,000.

Another error, which does not affect the results but may decrease efficiency of the optimization process, is that constraint (7) is always an unbinding constraint and hence useless in the model. This is because constraint (2) guarantees that $\left(\sum_{r \in R: C_r = c} x_r\right)$ must always be greater than or equal to one, therefore constraint (7) becomes $y_{ec} \le 1$ which is a trivial constraint as y_{ec} is a binary decision variable. So, it is unnecessary to include this constraint in the formulation. Moreover, constraint (2) is a redundant constraint from a mathematical point of view. It ensures that at least one rule per class label is selected whereas the objective function (1) aims to select rules to maximize the total number of corrected classification regardless of class labels. Therefore, because of the objective function, constraint (2) is almost always an unbinding constraint. The only exception is the case where keeping even one rule from a particular class label is not in favor of the objective function. This happens when a class label has very few samples in the training set and all of the rules generated for that class label have conflicts with the other rules such that keeping even one of them diminishes predictive accuracy of the model. In such a case, this constraint becomes binding and one of those rules is selected, though either the accuracy or the coverage of the classifier decreases as the result. However, some data sets need at least one rule per class label. For example, those in image processing or imbalanced data sets where some classes have few samples. This constraint can be removed from the model if the data set under consideration does not need this consideration.

3. Experimental study

To evaluate the proposed corrections on the original model, we repeated the experimental study carried out in [1] on the same data sets obtained from the UCI repository of machine learning databases[1]. Aydogan et al. [1] did not report the weights they used in their experiments for the fitness function of the GA. However, they provided us these weights in an email correspondence. These weights are $W_1 = (0.8, 0.05, 0.05, 0.1)$. We coded the original algorithm as it was described in [1] with the provided weights and termed it O-hGA$_r$. The results of this model are used as a baseline to analyze the corrected formulation. However, our experimental study showed that the algorithm performs better with $W_2 = (0.6, 0.05, 0.05, 0.3)$. We tested the corrected algorithm with both W_1 and W_2 and termed them C-hGA$_{W1}$ and C-hGA$_{W2}$, respectively. All models were coded in ILOG CPLEX 12.6 Java API and ran on a workstation equipped with an Intel Xeon CPU W3520 (2.67GHz) and 48 GB of RAM memory.

A 10-fold cross validation model was used to test the algorithms. Since all algorithms in this study are non-deterministic, we run each data set three times with different seed numbers to reduce the effect of randomness. The average results for 30 runs were reported for each data set. Table 1 presents the average accuracy of the original hGA and the corrected hGA algorithms in the training (%Tra) and test sets (%Tst), the average number of rules (#Rul), and the average computational time (Tim) in seconds, respectively. The original results reported in [1] were copied directly to Table 1 and termed O-hGA. Due to computational complexity, C-hGA$_{W2}$ may not find an optimal solution for the large data sets in a reasonable time. We tested different termination criteria and found out that setting a one hour termination condition on the optimization process is an efficient criterion in terms of the quality of the solution and the computational resource. Thus, C-hGA$_{W2}$ terminates the optimization process once the optimization process reaches one hour search without finding an optimal solution and reports the best obtained solution. The results obtained by this condition are marked in Table 1.

The average accuracy reported for the O-hGA [1] on the test sets in the original paper are much higher than the ones obtained by the O-hGA$_r$ for all data sets. This is true for the training sets as well except that O-hGA$_r$ obtained higher accuracy in Libras Mov. We cannot explain the large differences between the results reported in the original paper [1] and our results with the same weights.

C-hGA$_{W1}$ obtained higher accuracy than the O-hGA$_r$ in training and test sets for all data sets. It shows that the corrected formulation improves the accuracy of the original algorithm.

Comparing predictive accuracy between C-hGA$_{W2}$ and C-hGA$_{W1}$ shows that the former model obtained higher predictive accuracy on both test and training sets in almost all data sets except the

[1] http://archive.ics.uci.edu/ml/

Table 1
Accuracy results of original hGA and corrected hGA.

Data set	Corrected formulation													
	O-hGA [1]		O-hGA$_r$				C-hGA$_{W1}$				C-hGA$_{W2}$			
	%Tra	%Tst	%Tra	%Tst	#Rul	Tim	%Tra	%Tst	#Rul	Tim	%Tra	%Tst	#Rul	Tim
Bupa	68.91	63.82	5.21	4.16	87.20	<1	7.30	4.83	99.33	<1	60.00[a]	56.33	25.37	389
Cleveland	63.02	65.86	31.90	23.79	199.47	<1	52.14	34.12	249.53	<1	69.07	50.39	115.03	<1
Ecoli	88.98	89.09	45.15	43.25	29.70	<1	71.00	66.07	28.70	<1	77.90	71.63	53.40	<1
Glass	69.22	73.33	26.83	22.12	277.00	<1	36.24	28.04	298.77	<1	56.99	46.73	147.20	<1
HillValley1	52.93	52.48	1.88	0.96	70.37	32	3.61	2.09	98.13	32	42.42[a]	42.30	84.03	280
HillValley2	52.29	52.37	2.26	1.65	76.47	32	3.84	2.61	99.30	32	44.76	42.99	84.57	240
Iris	98.00	98.00	69.95	69.56	132.8	<1	88.96	87.33	144.37	<1	94.10	92.67	97.63	<1
Libras Mov.	64.94	65.56	68.92	51.30	676.57	72	82.71	58.33	724.00	60	85.16	61.30	689.27	65
Page-blocks	96.30	95.83	78.67	78.32	36.23	18	90.90	90.75	39.27	12	91.31	91.19	45.90	25
Parkinsons	91.88	96.32	58.10	57.61	68.87	<1	76.41	72.99	84.70	<1	89.19	83.59	11.30	<1
Pen-based	85.69	85.07	33.82	33.63	295.40	120	59.09	58.45	439.77	95	73.06	72.17	252.50	120
Ringnorm	92.50	93.10	6.09	5.84	55.90	25	16.48	15.64	97.60	19	83.37[a]	83.03	33.07	3600
Sonar	91.22	92.50	47.49	40.71	78.47	2	66.45	53.52	94.17	2	80.84	65.71	58.53	<1
Twonorm	91.10	90.70	3.40	3.43	47.10	21	14.75	14.48	99.10	16	85.22[a]	84.74	34.63	3600
Wdbc	95.44	95.36	56.48	56.36	23.07	3	93.10	92.38	52.87	2	94.94	93.38	21.83	60
Wine	98.88	98.82	84.21	82.59	69.97	<1	97.09	93.26	78.10	<1	98.00	90.26	16.73	<1
Yeast	51.35	50.90	2.56	1.57	156.60	5	11.17	8.72	123.80	3	15.95[a]	13.01	180.90	2400
Average	83.02	74.83	36.64	33.93	140.07	19.41	51.25	46.09	167.74	15.99	73.08	67.15	114.82	641.50

[a] Optimization terminated for the MIP once the elapsed time reached one hour.

Table 2
Wilcoxon's test to analyze significant difference between results on the test sets, $\alpha = 0.05$.

Pair-wise comparisons	W^+	W^-	p-value	Hypothesis
O-hGA [1] vs. O-hGA$_r$	153	0	0.0003	Rejected
O-hGA$_r$ vs. C-hGA$_{W1}$	0	153	0.0003	Rejected
O-hGA$_r$ vs. C-hGA$_{W2}$	0	153	0.0003	Rejected
C-hGA$_{W2}$ vs. C-hGA$_{W1}$	149	4	0.0006	Rejected

Wine data set in which C-hGA$_{W1}$ obtained higher accuracy in the test set.

We utilized non-parametric statistical tests to show significant differences among the results. We used Wilcoxon's Signed-Rank test [2] for pair-wise comparisons, and Friedman's test [3] and Iman and Davenport's test [4] for multiple comparison. Table 2 shows the results of the pair-wise comparisons between O-hGA, O-hGA$_r$, C-hGA$_{W1}$ and C-hGA$_{W2}$ using the Wilcoxon's Signed-Rank test. Wilcoxon's test detects significant differences between O-hGA and O-hGA$_r$, and between O-hGA$_r$ and C-hGA$_{W1}$. The results of the test and the higher predictive accuracy obtained by C-hGA$_{W1}$ show that the corrected formulation clearly improves the original model. Wilcoxon's test also detects significant differences between C-hGA$_{W2}$ and C-hGA$_{W1}$. Considering the higher predictive accuracy obtained by C-hGA$_{W2}$, we can conclude that it significantly outperforms C-hGA$_{W1}$.

Table 3
Statistical analysis with Friedman's and Iman-Davenport's tests among the three experiments on the test sets, $\alpha = 0.05$.

Test	Statistics	Critical value	p-value	Hypothesis
Friedman	32.12	5.99	<0.0000	Rejected
Iman-Davenport	273.36	3.29	<0.0000	Rejected

Table 3 summarizes Friedman's and Iman-Davenport's tests on O-hGA$_r$, C-hGA$_{W1}$ and C-hGA$_{W2}$. Both statistical tests reject equivalence of results between the considered algorithms.

References

[1] E.K. Aydogan, I. Karaoglan, P.M. Pardalos, hGA: Hybrid genetic algorithm in fuzzy rule-based classification systems for high-dimensional problems, Appl. Soft Comput. 12 (2) (2012) 800–806.
[2] F. Wilcoxon, Individual comparisons by ranking methods, Biomet. Bull. 1 (6) (1945) 80–83.
[3] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Am. Stat. Assoc. 32 (200) (1937) 675–701.
[4] R.L. Iman, J.M. Davenport, Approximations of the critical region of the friedman statistic, Commun. Stat.-Theory Methods 9 (6) (1980) 571–595.