AN EVOLUTIONARY APPROACH FOR LEARNING RULE WEIGHTS IN FUZZY RULE-BASED CLASSIFICATION SYSTEMS

Fahimeh Farahbod¹ and Mahdi Eftekhari²

¹Department of Computer Engineering, Shahid Bahonar University, Kerman, Iran
f.farahbod@eng.uk.ac.ir

²Department of Computer Engineering, Shahid Bahonar University, Kerman, Iran
m.eftekhari@uk.ac.ir

ABSTRACT

Rule weights often have been used to improve the classification accuracy without changing the position of antecedent fuzzy sets. Recently, fuzzy versions of confidence and support merits from the field of data mining have been widely used for rules weighting in fuzzy rule based classifiers. This paper proposes an evolutionary approach for learning rule weights and uses more flexible equations, which are evolved by genetic network programming. We perform some experiments using 15 well-known UCI data sets to examine efficiency of these novel equations; then, to analysis the results obtained in this experiments, a nonparametric statistical test is used. The results show that the performance of the fuzzy rule-based classification systems can be improved by using this proposed method for rule weight specification.

KEYWORDS

Pattern Classification, Fuzzy Systems, Rule Weighting, Genetic Network Programming

1. INTRODUCTION

In recent years, fuzzy models have been used widely because they are able to work with imprecise data and acquired knowledge with these models is more interpretable than the black-box models. Fuzzy models are able to handle the complex nonlinear problems. The fuzzy modelling process has generally to deal with an important trade-off between the accuracy and the interpretability of the model. Recently, tendency to look for a good balance between the accuracy and the interpretability has increased. Fuzzy Rule-Based Classification System (FRBCS) is a special case of fuzzy modelling. FRBCS focuses on finding a compact set of fuzzy if-then classification rules to model the input-output behaviour of the system. The input of the FRBCS is a number of pre-labelled classification examples, and the output of this system is a crisp and discrete value. One important advantage of a FRBCS is its interpretability.

Different methods that have been used to improve the accuracy of FRBCS can be grouped into two main categories. The first category contains methods for generating and adjusting antecedent fuzzy sets from numerical data, and the second group comprises methods in which expert knowledge is used to build FRBCS and rule weighting is employed to improve the accuracy of FRBCS. The second methods are better, because adjusting weight of rules is much easier than the
adjusting antecedent fuzzy sets and classification performance can be improved without changing the position of fuzzy sets and parameters of fuzzy sets given by domain experts. Many approaches have been proposed for learning fuzzy if-then rules from numerical input data for classification problems. Below are mentioned some of them.

Ishibuchi and Yamamoto in [1] proposed two heuristic methods for rule weight specification and compared four heuristic specification methods of rule weights with one another. The definitions of these heuristic methods are given in Section 3. Nakashima and et al. in [2] proposed a fuzzy rule-based classification system that allows the incorporation of weighted training patterns which can be used to adjust the sensitivity of the classification with respect to certain classes. They re-formulated the pattern classification problem as a cost minimization problem, and used the cost of misclassification or rejection of each pattern as the weight of it. Chen and Wang in [3] exhibited the connection between fuzzy classifiers and kernel machines, and proposed a support vector learning approach to construct fuzzy classifiers so that a fuzzy classifier can have good generalization ability in a high dimensional feature space. Sánchez and et al. in [4] combined genetic programming operators with simulated annealing to search the best rules. They used a simulated annealing-based method for inducing structure of a fuzzy classifier, and used macromutation operator from tree-shaped genotype genetic algorithms as adjacency operator. Jesus and et al. in [5] proposed a novel adaboost algorithm to learn fuzzy-rule-based classifiers. They applied an evolutionary boosting scheme to approximate and descriptive fuzzy-rule bases. Otero and Sánchez in [6] applied Logitboost to learn fuzzy rules in classification problems. Sánchez and Otero in [7] proposed a boosting-based genetic method to learn weighted fuzzy rules. González and Pérez in [8] proposed some modifications of the genetic algorithm of the SLAVE learning algorithm, including a feature selection model to select the appropriate features for a problem. This modification dynamically explores the set of possible variables in order to find the most useful rule and the most interesting variables for this rule. Structural learning algorithm on vague environment (SLAVE) is a genetic learning algorithm that uses the iterative approach to learn fuzzy rules. Ishibuchi and et al. in [9] combined two fuzzy genetics-based machine learning approaches (i.e., Michigan and Pittsburgh) into a single hybrid algorithm. Their hybrid algorithm is based on the Pittsburgh approach where a set of fuzzy rules is handled as an individual. Genetic operations for generating new fuzzy rules in the Michigan approach are utilized as a kind of heuristic mutation for partially modifying each rule set. Mansoori and et al. in [10] proposed a novel steady-state genetic algorithm to extract a compact set of good fuzzy rules from numerical data (SGERD). Zolghadri and Mansoori in [11] proposed a method of fuzzy rule weight specification. They used 2-class receiver operating characteristic (ROC) analysis to find the best threshold (resulting in maximum classification accuracy) for each rule in rule-base. This threshold is used as the weight of the rule. Zolghadri and Taheri in [12] proposed a method of learning rule weights in fuzzy rule-based classification systems. This method is a hill-climbing search method. The method starts with an initial solution (an initial rule-base with initial weights) and sequentially improves the solution by finding a neighbour solution that is better than the current one. A neighbour solution is different in the value of just one parameter (the weight of one rule) compared with the current solution.

Genetic programming (GP) is an evolutionary algorithm-based methodology inspired by biological evolution to find computer programs that perform a user-defined task. Genetic Network Programming (GNP) is an extension of GP, which is an evolutionary computation method with directed graphs as genes, and could be applied to many areas due to its compact structure. The basic structure of GNP genome structure is networks. So this can reduce the searching space for the solutions that leads to good performance than GP. We propose an evolutionary approach for learning rule weights, which is evolved by genetic network programming.
The paper is organized as follows. In Section 2, in a nutshell the method used for designing FRBCS from numerical data is presented. In Section 3, four ordinary heuristic method of rule weight specification are given. In Section 4, the proposed method is explained. In Section 5, we present the experiment results. Finally, we draw conclusions in Section 6.

2. GENERAL DESIGN OF FUZZY RULE-BASED CLASSIFICATION SYSTEM

FRBCS is composed of three principal components: database, rule-base and reasoning method. The database contains the fuzzy set definitions related to the linguistic terms used in the fuzzy rules. The rule base consists of a set of fuzzy if-then rules in the form of "if a set of conditions are satisfied, then a set of consequences can be inferred". Reasoning method uses information from database and rule-base to determine a class label for patterns and to classify them.

In this work, we use a simple and efficient heuristic method for constructing FRBCS. Let us assume that our pattern classification problem is an \( n \)-dimensional problem with \( C \) classes and \( m \) training patterns, \( X_p = \{x_{p1}, x_{p2}, \ldots, x_{pn}\}, p = 1, 2, \ldots, m \). Usually, each attribute of the given training patterns is normalized into a unit interval \([0, 1]\) by using a linear transformation that preserves the distribution of training patterns. We used 14 fuzzy sets showed in Fig. 1 similar to Ref. [1] to partition the domain interval of each input attribute. Triangular shaped fuzzy sets are used, because they are simple and more human understandable.

Let us assume \( X_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\} \) is the input attribute vector, \( R_q \) is the label of the \( q \)-th fuzzy if-then rule, \( A_{q1}, A_{q2}, \ldots, A_{qn} \) are antecedent fuzzy sets on the unit interval \([0, 1]\), \( C_q \) is the consequent class, \( CF_q \) is the certainty grade of rule \( R_q \). We use fuzzy rules of following type:

\[
\text{if } x_{i1} \text{ is } A_{q1} \text{ and } \ldots \text{ and } x_{in} \text{ is } A_{qn} \text{ then } C_q \text{ with } CF_q. 
\]

In order to classify an input pattern \( X_p = \{x_{p1}, x_{p2}, \ldots, x_{pn}\} \), the compatibility degree of the pattern with each rule is calculated. In case of using product as \( T \)-norm operator to model the “and” connectives in the rule antecedent, the compatibility degree of pattern \( X_p \) with the rule \( R_q \) can be calculated as follows:

\[
\mu_q(X_p) = \prod_{i=1}^{n} \mu_{q_i}(X_{pi}). 
\]  (1)

Where \( \mu_{q_i}(\cdot) \) is the membership function of the antecedent fuzzy set \( A_{q_i} \). Assume \( C_q \) is a class label for \( r \) patterns, confidence (denoted by Conf), support (denoted by Supp), and lift (denoted by Lift) of a fuzzy rule are defined as follows:

$$Conf(A_q \Rightarrow Class C_q) = \frac{\sum_{p \in Class C_q} \mu_q(X_p)}{\sum_{p=1}^{m} \mu_q(X_p)}, \quad (2)$$

$$Supp(A_q \Rightarrow Class C_q) = \frac{1}{m} \cdot \sum_{X_p \in Class C_q} \mu_q(X_p), \quad (3)$$

$$Lift(A_q \Rightarrow Class C_q) = \frac{\sum_{X_p \in Class C_q} \mu_q(X_p)}{\sum_{p=1}^{m} \mu_q(X_p)} / \frac{r}{m}. \quad (4)$$

The most common reasoning methods are single winner reasoning method and weighted vote reasoning method. In the case of using single winner reasoning method for classifying new patterns (assume the classifier have $R$ rules), the single winner rule $R_w$ is determined as follows:

$$\mu_w(X_p).CF_w = \max \{\mu_q(X_p).CF_q : q = 1, ..., R\} \quad (5)$$

$$w = \arg \max_q \{\mu_q(X_p).CF_q : q = 1, ..., R\} \quad (6)$$

We are generated fuzzy rules with two antecedent conditions and product of confidence and support of rule is used as a certainty grade of the rule. The consequent class of an antecedent combination is specified by finding the class with maximum product of confidence and support. When the consequent class cannot be uniquely determined, the rule is not generated. We are used an evolutionary approach to specify rule weights.

The new pattern $X_p$ is classified as class $C_w$, which is the consequent class of the winner rule $R_w$. If no fuzzy rule covers the $X_p$ and compatible with it or if for $X_p$ multiple fuzzy rules have the same maximum value (product of compatibility grade and certainty grade), but different consequent classes, the classification of $X_p$ is rejected.

3. HEURISTIC RULE WEIGHT SPECIFICATION METHODS

In order to assign a weight to each fuzzy classification rule, several heuristic measures proposed in past researches, four of them will be explained here. One definition of the rule weight specification proposed in [1] is as follows:

$$CF_q^{I} = conf(A_q \Rightarrow Class C_q) \quad (7)$$

The second definition of rule weight specification proposed in [1] is as follows:

$$CF_q^{II} = conf(A_q \Rightarrow Class_q) - conf_{Ave} \quad (8)$$

Where, $conf_{Ave}$ is the average confidence of the fuzzy rules having $A_q$ in antecedent part and consequent classes are not $Class_q$. For a $C$-class problem, $conf_{Ave}$ can be calculated as follows:

$$conf_{Ave} = \frac{1}{C-1} \cdot \sum_{b=1, b\neq Class_q}^{C} conf(A_q \Rightarrow Class_b) \quad (9)$$

The third definition of rule weight specification proposed in [1] is as follows:

$$CF_q^{III} = conf(A_q \Rightarrow Class_q) - conf_{2nd} \quad (10)$$
Where, $conf_{2nd}$ is the largest confidence of the fuzzy rules having $A_q$ in antecedent part and consequent classes are not $Class_q$. For a C-class problem, $conf_{2nd}$ can be calculated as follows:

$$conf_{2nd} = \max \{conf(A_q \Rightarrow Class_h) | h = 1,2,\ldots,C; h \neq Class_q \}$$ (11)

The fourth definition of rule weight specification proposed in [1] is as follows:

$$CF_{q}^{IV} = conf(A_q \Rightarrow Class_q) - conf_{sum}$$ (12)

Where, $conf_{sum}$ is the sum of confidence of the fuzzy rules having $A_q$ in antecedent part and consequent classes are not $Class_q$. For a C-class problem, $conf_{sum}$ can be calculated as follows:

$$conf_{sum} = \sum_{h=1,h \neq Class_q}^{C} conf(A_q \Rightarrow Class_h).$$ (13)

4. **Evolvreny Approach for Rule Weighting**

In this section, we explain the proposed evolutionary method, which is evolved by GNP and for rule weight specification is used.

4.1. **GNP structure**

The general gene structure of the GNP individual was inspired from the research in [13]. The basic structure of this GNP individual consists of one memory structure and three kinds of nodes: start node, judgement node and processing node. The start node indicates where the transition starts, the judgement nodes decide direction of transition and the processing nodes generate the equations which are used for rule weight specification.

Each processing node is connected to one judgement node and is associated with several components in memory, such as one or two weights in weight memory, one or two measures in measure memory, one operator in operator memory and one or several equations in equation memory. Each judgement node is connected to one judgement node and several processing nodes. After executing a judgement node, one of the processing nodes or other judgement node, those are connected to this judgement node, is selected as the next node to be executed.

Memory structure is associated with each processing node and consists of four parts: weight memory, measure memory, operator memory and equation memory. Weight memory consists of real numbers between 0 and 1, which have been used as parameters in generating equations. Measure memory consists of confidences, supports and lifts of rules, which have been used as operands in generating equations, $M = \{Conf, Supp, Lift\}$. Operator memory consists of the set of operators used to combine parameters and operands for generating equations, $O = \{+,-,\times,\div, SQ, SR, MAX, MIN, ABS\}$. Where ABS is the abbreviation of absolute value, SQ is the abbreviation of square and SR is the abbreviation of square root. Equations generated by GNP are stored into the equation memory. Once the equation is generated, we could use it to calculate weight value for each rule. The equations stored in equation memory could be used as operand to generate more complex equations.

In each GNP individual, various combinations of measures, equations and operators could be generated, each of which will be used to specify classification rule weights and gives classification accuracy independently. The fitness value of the individual is the best accuracy.
Crossover, mutation of start node, mutation of judgement node, and mutation of processing node are genetic operators used here for evolving GNP, which are illustrated by Fig. 2. The general gene structure of GNP individual used is illustrated by Fig. 3. This GNP individual have one start node (S), two judgement nodes (J₁ and J₂), and two processing nodes (P₁ and P₂). S is connected to J₁, J₁ is connected to P₁ and P₂, P₁ is connected to J₂, J₂ is connected to P₁ and P₂, and P₂ is connected to J₁. After executing S, J₁ will be the next node. After executing J₁, P₁ or J₂ may be selected as a next node. We may select P₁. P₁ is associated with two weight values w₁ and w₂, a measure m₁, an operator o₁, and an equation e₁. After P₁ is executed, a new equation, i.e., e₂, is generated as e₂ = w₁ × m₁ × o₁ × w₂ × e₁, where w₁ and w₂ are real numbers between interval [0, 1], m₁ is Conf or Supp or Lift, o₁ could be +, −, ×, ÷, MAX or MIN, and e₁ is an equation generated in past transitions. The following nodes are executed in the same manner. We stop the transition process after a certain number of transition steps (we import a parameter named maximum transition steps).
4.2. Proposed algorithm

The flowchart of the proposed algorithm, we are used for designing FRBCS and weighting rules is shown in Fig. 4. In part 1, after fuzzy rules are generated, we use an equation which is evolved by GNP to specify weight fuzzy rules and build a FRBCS, then test this system on testing data. In part 2, after some initial GNP individual are generated, we calculate fitness value for each individual and use evolutionary operators to generate a good population. This process is terminated when a certain number of generations are executed and a certain number of individual evaluated. Each individual usually could generate more than one equation; we set the highest accuracy as the fitness value of the individual. The equation of individual with highest fitness value is selected as the best equation. In part 3, for evaluation the performance of an equation generated, we build a FRBCS and use this equation to specify rule weights, and then test this system on training data. The data for testing here is the training data, since all the steps of part 3 are within the training phase of part 1.
5. EXPERIMENT RESULTS

In this section, we evaluate the performance of the new proposed rule-weighting method. We have used 15 data sets with numerical attributes from the University of California, Irvine machine learning repository (UCI) [15], all of them valid for classification tasks. Table 1 shows specification of these data sets. For each data set, the name, number of samples, number of attributes and number of classes are given. The parameters setting for GNP in the experiments are shown in Table 2. Since the speed is one of the most important considerations of classification problems, we select the small size GNP to limit the search space and do not spend too much time on evaluation.

Table 1. Statistics of data sets used in this paper.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Number of attributes</th>
<th>Number of samples</th>
<th>Number of classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wisconsin (Breast cancer wisconsin)</td>
<td>10</td>
<td>699</td>
<td>2</td>
</tr>
<tr>
<td>Pima (Pima diabetes)</td>
<td>8</td>
<td>768</td>
<td>2</td>
</tr>
<tr>
<td>Haberman</td>
<td>4</td>
<td>306</td>
<td>2</td>
</tr>
<tr>
<td>Heart Statlog</td>
<td>13</td>
<td>270</td>
<td>2</td>
</tr>
<tr>
<td>Liver (Liver disorders)</td>
<td>7</td>
<td>345</td>
<td>2</td>
</tr>
<tr>
<td>Labor</td>
<td>16</td>
<td>57</td>
<td>2</td>
</tr>
<tr>
<td>Wine</td>
<td>13</td>
<td>178</td>
<td>3</td>
</tr>
<tr>
<td>Thyroid (New Thyroid)</td>
<td>5</td>
<td>215</td>
<td>3</td>
</tr>
<tr>
<td>Balance (Balance-Scale)</td>
<td>4</td>
<td>625</td>
<td>3</td>
</tr>
<tr>
<td>Iris</td>
<td>5</td>
<td>150</td>
<td>3</td>
</tr>
</tbody>
</table>
For generating the best equation (which is used to specify rule weights) for each data set, we have employed the ten-fold cross-validation (10-CV) testing method as a validation scheme to perform the experiments and analyze the results. We have run the algorithms five times in order to obtain a sample of 50 results, which are averaged, for each data set. In ten-fold cross-validation method, each data set is randomly divided into ten disjoint sets of equal size (the size of each set is \( m / 10 \), where \( m \) is the total number of patterns in data set). The FRBCS is trained ten times, each time one of ten sets hold out as a test set for evaluating FRBCS and the nine remainder sets are used for training. The classification accuracy is computed in each time and the estimated classifier performance is the average of these 50 classification accuracies.

The experiment results are listed in Table 3. In this table, the first column shows the names of data sets. The second column gives the average classification accuracy (obtained from the 10-CV testing method) of the FRBCS, which used proposed new rule weighting method. The 3\(^{\text{th}}\), 4\(^{\text{th}}\), 5\(^{\text{th}}\), and 6\(^{\text{th}}\) columns, gives the mean absolute error (MAE), root mean squared error (RMSE), relative absolute error (RAE), and root relative squared error (RRSE), respectively. There are 50 runs for each data set, and we record the best accuracy and error measures for each dataset. The best equations (for each data set) obtained in experiments are shown in Table 4. Let us assume that our pattern classification problem has \( m \) patterns, \( p_i \) is the predicted class label for pattern with classifier \( a \) and \( t_i \) is the target class label for pattern. The definitions of accuracy and error measures are as follows, respectively.

\[
\text{accuracy} = \frac{m_c}{m} \tag{14}
\]

Where, \( m \) is the total number of patterns and \( m_c \) is the number of correctly classified patterns.

\[
\text{MAE} = \frac{1}{m} \sum_{i=1}^{n} |p_i - t_i| \tag{15}
\]

\[
\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (p_i - t_i)^2}{m}} \tag{16}
\]
By the proposed method, we could generate various combinations of operators and measures and each combination could be used to specify rule weights. For comparing classification accuracy of proposed weighting method with other weighting methods introduced in Section 3, we select one equation and use it to specify rule weights (for 9 data sets in Table 5). This equation is as follows:

Equation: \( 0.09 \times SR \left( \text{ABS} \left( (0.81 \times \text{Supp}) - (0.59 \times \text{Conf}) \right) \right). \) (19)

Table 5 shows these comparing results, the best results in each row (for each data set) are highlighted by boldface. We have employed the FT-FT testing method to perform these experiments and analyze the results. In Full Train–Full Test (FT–FT), the full data set is used in training and testing phases, in training phase to construct the rule-base and in the testing phase to evaluate rule-base.

Experimental analysis of the performance of a proposed method is a necessary task in an investigation. For the sake of comparison, Table 7 gives the accuracies obtained by the proposed method as well as those of previously developed methods (especially fuzzy rule learning and evolutionary fuzzy rule learning approaches) over different data sets. In this table, we have used 9 data sets with numerical attributes from the University of California, Irvine machine learning repository (UCI) [15], all of them valid for classification tasks. We are measured the performance of each classifier by means of its accuracy in test data by using 5 repetitions of 10-CV cross-validation. The best results in each row (for each data set) are highlighted by boldface. In the Table 7, the first column shows the names of datasets. The average classification accuracy for each data set by the algorithms are introduced in Table 6, are showed in 2th, 3th, 4th, 5th, 6th, 7th, 8th columns, resp.ectively. At last, average classification accuracy for each data set by the algorithm is proposed in this paper is showed in 9th column. Experimental results in Table 7 show that the proposed method achieves a higher average classification accuracy rate in vast majority of experiment cases. However, this observation-based evaluation does not reflect whether or not the differences among the methods are significant.

We are used statistical tests to make sure that the difference is significant, that is, big enough that it could not have happened by chance, or in other words, very unlikely to have been caused by chance - the so-called p-value of the test [16]. To evaluate the performance of the proposed method, we are used Friedman test [17], which is a non-parametric statistical analysis based on multiple comparison procedures. In order to perform a multiple comparison, it is necessary to check whether all the results obtained by the algorithms present any inequality. Friedman test, ranks the algorithms for each data set separately, the best performing algorithm getting the rank of 1, the second best rank 2, and so on. In case of ties, average ranks are assigned. Under the null-hypothesis, it states that all the algorithms are equivalent, so a rejection of this hypothesis implies the existence of differences among the performance of all the algorithms studied [18]. Friedman’s test’s way of working is described as follows.
Let \( r_i^j \) be the rank of the \( j \)-th of \( k \) algorithms on the \( i \)-th of \( N \) data sets. The Friedman test compares the average ranks of algorithms, \( R_j = \frac{1}{N} \sum r_i^j \). Under the null-hypothesis, which states that all the algorithms are equivalent and so their ranks \( R_j \) should be equal, the Friedman statistic is distributed according to \( \chi^2 \) with \( k - 1 \) degrees of freedom and is as follows [19]:

\[
\chi_F^2 = \frac{12N}{k(k+1)} \left[ \sum_j R_j^2 - \frac{k(k+1)^2}{4} \right].
\]

(20)

Average ranks obtained by each method in the Friedman test are shown in Table 8. In this table, the value of Friedman statistic (distributed according to chi-square with 7 degrees of freedom) is 34.305556 and \( p \)-value computed by this test is 0.000015. These rank values will be useful to calculate the \( p \)-values and to detect significant differences between the methods. Evidently, the rank assigned to proposed method is less than other method ranks. Hence, proposed method is the best performing method.

After one statistical test used, a post-hoc test could be used in order to find whether the control method, which is the proposed method presents statistical differences with regard to the remaining methods in the comparison [18]. The post-hoc methods we used are Holm’s test and Finner test. Holm’s test [20] is a multiple comparison procedure that can work with a control algorithm (which is usually the best according to Friedman rankings computation) and compares it with the remaining methods. The test statistics for comparing the \( i \)-th and \( j \)-th method using this procedure is as follows:

\[
z = \frac{(R_i - R_j)}{\sqrt{k(k+1)/6N}}.
\]

(21)

The \( z \) value is used to find the corresponding probability from the table of normal distribution, which is then compared with an appropriate level of confidence \( \alpha \) [18]. Holm’s test adjusts the value for \( \alpha \) in order to compensate for multiple comparisons.

Holm’s test adjusts the value of \( \alpha \) in a step-down manner. Let \( p_1, p_2, ..., p_{k-1} \) be the ordered \( p \)-values (smallest to largest), so that \( p_1 \leq p_2 \leq ... \leq p_{k-1} \), \( H_1, H_2, ..., H_{k-1} \) be the corresponding hypotheses. The Holm procedure rejects \( H_i \) to \( H_{i-1} \) if \( i \) is the smallest integer such that \( p_i > \alpha / (k-i) \). Holm’s step-down procedure starts with the most significant \( p \)-value. If \( p_1 \) is below \( \alpha / (k-1) \), the corresponding hypothesis is rejected and we are allowed to compare \( p_2 \) with \( \alpha / (k-2) \). If the second hypothesis is rejected, the test proceeds with the third, and so on. As soon as a certain null hypothesis cannot be rejected, all the remaining hypotheses are retained as well [18].

The Finner procedure [21] adjusts the value of \( \alpha \) in a step-down manner, as Holm’s method do. It rejects \( H_i \) to \( H_{i-1} \) if \( i \) is the smallest integer so that \( p_i > 1 - (1 - \alpha)^{(k-i)}/(k-i) \) [18].

The \( p \)-values obtained in by applying Holm method and Finner method as post hoc methods over the results of Friedman procedure are shown in Table 9. Holm’s procedure rejects those hypotheses that have a \( p \)-value \( \leq 0.025 \) and Finner’s procedure rejects those hypotheses that have a \( p \)-value \( \leq 0.05 \). As Table 9 shows, the procedure of Holm verifies that proposed method performs better than all other approaches except approach proposed in [2], because all approaches
except approach proposed in [2] have a Holm \( \leq 0.025 \), and the procedure of Finner verifies that proposed method is the best method, because all approaches have a Finner \( \leq 0.05 \).

After a post-hoc method used, the adjusted \( p \)-values method used for computing these exact \( p \)-values for each test procedure. The adjusted \( p \)-value for the Holm procedure is computed by \( p_{\text{Holm}} = (k - i)p_i \). Once all of them are computed for all hypotheses, it is not possible to find an adjusted \( p \)-value for the hypothesis \( i \) lower than for the hypothesis \( j, j < i \). In this case, the adjusted \( p \)-value for hypothesis \( i \), is set to the same value as the one associated to hypothesis \( j \) [22]. Adjusted \( p \)-values obtained are shown in Table 10.

Table 3. Experiment results (10-CV test method).

<table>
<thead>
<tr>
<th>Data set</th>
<th>Accuracy</th>
<th>MAE</th>
<th>RMSE</th>
<th>RAE</th>
<th>RRSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wisconsin</td>
<td>97.04</td>
<td>0.003</td>
<td>0.049</td>
<td>0.065</td>
<td>0.329</td>
</tr>
<tr>
<td>Pima</td>
<td>76.97</td>
<td>0.023</td>
<td>0.150</td>
<td>0.506</td>
<td>1.001</td>
</tr>
<tr>
<td>Haberman</td>
<td>96.66</td>
<td>0.013</td>
<td>0.036</td>
<td>0.066</td>
<td>0.113</td>
</tr>
<tr>
<td>Heart Statlog</td>
<td>82.45</td>
<td>0.017</td>
<td>0.127</td>
<td>0.356</td>
<td>0.826</td>
</tr>
<tr>
<td>Liver</td>
<td>65.19</td>
<td>0.034</td>
<td>0.186</td>
<td>0.713</td>
<td>1.191</td>
</tr>
<tr>
<td>Labor</td>
<td>86.30</td>
<td>0.008</td>
<td>0.058</td>
<td>0.279</td>
<td>0.456</td>
</tr>
<tr>
<td>Wine</td>
<td>96.07</td>
<td>0.003</td>
<td>0.052</td>
<td>0.062</td>
<td>0.213</td>
</tr>
<tr>
<td>Thyroid</td>
<td>95.98</td>
<td>0.006</td>
<td>0.064</td>
<td>0.092</td>
<td>0.276</td>
</tr>
<tr>
<td>Balance</td>
<td>89.58</td>
<td>0.013</td>
<td>0.142</td>
<td>0.153</td>
<td>0.472</td>
</tr>
<tr>
<td>Iris</td>
<td>96.66</td>
<td>0.046</td>
<td>0.031</td>
<td>0.044</td>
<td>0.114</td>
</tr>
<tr>
<td>Post</td>
<td>73.90</td>
<td>0.054</td>
<td>0.279</td>
<td>0.621</td>
<td>1.048</td>
</tr>
<tr>
<td>TAE</td>
<td>59.04</td>
<td>0.017</td>
<td>0.280</td>
<td>0.849</td>
<td>1.098</td>
</tr>
<tr>
<td>Car</td>
<td>86.63</td>
<td>0.070</td>
<td>0.156</td>
<td>0.293</td>
<td>0.664</td>
</tr>
<tr>
<td>Glass</td>
<td>63.46</td>
<td>0.035</td>
<td>0.442</td>
<td>0.421</td>
<td>0.686</td>
</tr>
<tr>
<td>Ecoli</td>
<td>84.47</td>
<td>0.033</td>
<td>0.286</td>
<td>0.285</td>
<td>0.636</td>
</tr>
</tbody>
</table>

Table 4. The best equations computed in experiments.

<table>
<thead>
<tr>
<th>Data set</th>
<th>The best equation obtained</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wisconsin</td>
<td>SQ(0.16×ABS(0.37×((0.07×conf)+(0.54×((0.25×supp)-(0.72×conf)))))</td>
</tr>
<tr>
<td>Pima</td>
<td>((0.32×Conf)-(0.23×Supp))×ABS((0.05×Lift)×(0.19×SQ((0.40×Conf)-(0.29×Supp))))</td>
</tr>
<tr>
<td>Haberman</td>
<td>(0.15×Supp)+(0.91×Conf)</td>
</tr>
<tr>
<td>Heart</td>
<td>MIN((0.91×Conf),(0.91×Lift))</td>
</tr>
<tr>
<td>Liver</td>
<td>(0.74×((0.91×Supp)-(0.75×Lift)))×(0.96×Conf)</td>
</tr>
<tr>
<td>Labor</td>
<td>MAX((0.56×Supp),(0.96×Conf))</td>
</tr>
<tr>
<td>Wine</td>
<td>(0.79×Conf)×(0.62×((0.01×Lift)-(0.21×Supp)))</td>
</tr>
<tr>
<td>Thyroid</td>
<td>SR(ABS((0.09×Conf)+((0.09×Supp)-(0.17×Lift×SQ(0.99×Supp))))×(0.49×Conf))</td>
</tr>
<tr>
<td>Balance</td>
<td>(0.98×MIN((0.86×ABS(0.12×Conf),(0.71×Supp)))-(0.54×Conf))</td>
</tr>
<tr>
<td>Iris</td>
<td>((0.07×Supp×Conf)+((0.02×SR(0.48×Conf×Lift)))×(0.05×SR(0.65×Conf×Lift))</td>
</tr>
<tr>
<td>Post</td>
<td>SR(0.43×Conf)</td>
</tr>
<tr>
<td>TAE</td>
<td>(0.17×SR(((0.82×SQ(0.55×Conf))-0.39×Lift)))</td>
</tr>
<tr>
<td>Car</td>
<td>ABS(0.4×((0.56×(0.21×Lift)+0.14×Supp))-(0.6×SQ(0.37×Lift)))×(0.04×Conf)</td>
</tr>
<tr>
<td>Glass</td>
<td>((0.08×Supp)-(0.16×Conf))×((0.01×conf)×(0.3×SR(0.57×Supp)))</td>
</tr>
<tr>
<td>Ecoli</td>
<td>(0.17×((0.34×Conf)×(0.78×Lift))+(0.21×Conf))-(0.46×Supp)</td>
</tr>
</tbody>
</table>
Table 5. Comparing the classification accuracy of proposed method with weighting method introduced in Section 3 (FT-FT test method).

<table>
<thead>
<tr>
<th>Data set</th>
<th>CF^I</th>
<th>CF^II</th>
<th>CF^III</th>
<th>CF^IV</th>
<th>Proposed method</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wisconsin</td>
<td>94.99</td>
<td>95.42</td>
<td>95.42</td>
<td>95.42</td>
<td>95.99</td>
<td>0.09 x SR (ABS ((0.81 x Supp) - (0.59 x Conf)))</td>
</tr>
<tr>
<td>Pima</td>
<td>67.69</td>
<td>71.22</td>
<td>71.22</td>
<td>71.22</td>
<td>77.86</td>
<td>0.09 x SR (ABS ((0.81 x Supp) - (0.59 x Conf)))</td>
</tr>
<tr>
<td>Haberman</td>
<td>74.18</td>
<td>74.18</td>
<td>74.18</td>
<td>74.18</td>
<td>75.49</td>
<td>0.09 x SR (ABS ((0.81 x Supp) - (0.59 x Conf)))</td>
</tr>
<tr>
<td>Liver</td>
<td>59.42</td>
<td>60.57</td>
<td>60.57</td>
<td>60.57</td>
<td>66.08</td>
<td>0.09 x SR (ABS ((0.81 x Supp) - (0.59 x Conf)))</td>
</tr>
<tr>
<td>Labor</td>
<td>94.73</td>
<td>94.73</td>
<td>94.73</td>
<td>94.73</td>
<td>98.24</td>
<td>0.09 x SR (ABS ((0.81 x Supp) - (0.59 x Conf)))</td>
</tr>
<tr>
<td>Thyroid</td>
<td>82.79</td>
<td>86.51</td>
<td>87.44</td>
<td>89.76</td>
<td>94.88</td>
<td>0.09 x SR (ABS ((0.81 x Supp) - (0.59 x Conf)))</td>
</tr>
<tr>
<td>Balance</td>
<td>99.31</td>
<td>99.31</td>
<td>99.31</td>
<td>99.31</td>
<td>100</td>
<td>0.09 x SR (ABS ((0.81 x Supp) - (0.59 x Conf)))</td>
</tr>
<tr>
<td>Glass</td>
<td>64.01</td>
<td>66.82</td>
<td>67.76</td>
<td>68.22</td>
<td>69.62</td>
<td>0.09 x SR (ABS ((0.81 x Supp) - (0.59 x Conf)))</td>
</tr>
<tr>
<td>Ecoli</td>
<td>66.96</td>
<td>67.55</td>
<td>73.21</td>
<td>77.08</td>
<td>82.14</td>
<td>0.09 x SR (ABS ((0.81 x Supp) - (0.59 x Conf)))</td>
</tr>
</tbody>
</table>

Table 6. Algorithms compared with proposed method in experiments.

<table>
<thead>
<tr>
<th>References</th>
<th>Authors</th>
<th>Year</th>
<th>Method name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Nakashima, Schaefer, Yokota and Ishibuchi</td>
<td>2007</td>
<td>M1</td>
</tr>
<tr>
<td>1</td>
<td>Ishibuchi and Yamamoto</td>
<td>2005</td>
<td>M2</td>
</tr>
<tr>
<td>9</td>
<td>Ishibuchi, Yamamoto and Nakashima</td>
<td>2005</td>
<td>M3</td>
</tr>
<tr>
<td>4</td>
<td>Sánchez and Couso</td>
<td>2001</td>
<td>M4</td>
</tr>
<tr>
<td>7</td>
<td>Sánchez and Otero</td>
<td>2007</td>
<td>M5</td>
</tr>
<tr>
<td>10</td>
<td>Mansoori, Zolghadri and Katebi</td>
<td>2008</td>
<td>M6</td>
</tr>
<tr>
<td>8</td>
<td>Gonzalez and Perez</td>
<td>2001</td>
<td>M7</td>
</tr>
</tbody>
</table>

Table 7. Comparing the classification accuracy of proposed method with some classification approaches (10-CV test method).

<table>
<thead>
<tr>
<th>Data set</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>M7</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ecoli</td>
<td>80.40</td>
<td>72.02</td>
<td>72.92</td>
<td>70.22</td>
<td>71.18</td>
<td>74.68</td>
<td>84.53</td>
<td>84.47</td>
</tr>
<tr>
<td>Glass</td>
<td>58.30</td>
<td>60.04</td>
<td>56.60</td>
<td>55.15</td>
<td>62.77</td>
<td>61.02</td>
<td>58.14</td>
<td>63.46</td>
</tr>
<tr>
<td>Haberman</td>
<td>73.80</td>
<td>73.19</td>
<td>72.19</td>
<td>71.56</td>
<td>73.81</td>
<td>73.52</td>
<td>71.90</td>
<td>96.66</td>
</tr>
<tr>
<td>Iris</td>
<td>96</td>
<td>92.66</td>
<td>96.66</td>
<td>93.33</td>
<td>93.33</td>
<td>95.33</td>
<td>95.33</td>
<td>96.66</td>
</tr>
<tr>
<td>Pima</td>
<td>73.71</td>
<td>73.06</td>
<td>74.09</td>
<td>72.93</td>
<td>75.01</td>
<td>73.05</td>
<td>73.32</td>
<td>76.97</td>
</tr>
<tr>
<td>Wine</td>
<td>94.01</td>
<td>93.82</td>
<td>71.01</td>
<td>89.24</td>
<td>94.41</td>
<td>92.09</td>
<td>92.12</td>
<td>96.07</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>96.85</td>
<td>91.12</td>
<td>95.70</td>
<td>95.56</td>
<td>95.41</td>
<td>93.55</td>
<td>95.42</td>
<td>97.04</td>
</tr>
<tr>
<td>Thyroid</td>
<td>97.25</td>
<td>82.24</td>
<td>93.54</td>
<td>92.57</td>
<td>92.61</td>
<td>87.03</td>
<td>90.75</td>
<td>95.98</td>
</tr>
<tr>
<td>Liver</td>
<td>59.37</td>
<td>57.87</td>
<td>64.95</td>
<td>57.37</td>
<td>64.63</td>
<td>57.34</td>
<td>59.06</td>
<td>65.19</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Ranking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M1</td>
<td>3.1111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M2</td>
<td>6.1111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3</td>
<td>4.1667</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M4</td>
<td>6.9444</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M5</td>
<td>3.8333</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M6</td>
<td>5.6111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M7</td>
<td>4.9444</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed method</td>
<td>1.2778</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8. Average rankings of algorithms by Friedman procedure.

Table 9. Post Hoc comparison table for $\alpha = 0.05$ (Friedman).

<table>
<thead>
<tr>
<th>$i$</th>
<th>Algorithm</th>
<th>$z$</th>
<th>$p$</th>
<th>Holm</th>
<th>Finner</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>M4</td>
<td>4.907477</td>
<td>0.000001</td>
<td>0.007143</td>
<td>0.007301</td>
</tr>
<tr>
<td>6</td>
<td>M2</td>
<td>4.185789</td>
<td>0.000028</td>
<td>0.008333</td>
<td>0.014548</td>
</tr>
<tr>
<td>5</td>
<td>M6</td>
<td>3.752777</td>
<td>0.00175</td>
<td>0.01</td>
<td>0.021743</td>
</tr>
<tr>
<td>4</td>
<td>M7</td>
<td>3.175426</td>
<td>0.001496</td>
<td>0.0125</td>
<td>0.028885</td>
</tr>
<tr>
<td>3</td>
<td>M3</td>
<td>2.501851</td>
<td>0.012355</td>
<td>0.016667</td>
<td>0.035975</td>
</tr>
<tr>
<td>2</td>
<td>M5</td>
<td>2.213176</td>
<td>0.026886</td>
<td>0.025</td>
<td>0.043013</td>
</tr>
<tr>
<td>1</td>
<td>M1</td>
<td>1.587713</td>
<td>0.112351</td>
<td>0.05</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 10. Adjusted P-values (Friedman), proposed method is the control algorithm.

<table>
<thead>
<tr>
<th>$i$</th>
<th>Algorithm</th>
<th>unadjusted $p$</th>
<th>$p_{\text{Holm}}$</th>
<th>$p_{\text{Finner}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M4</td>
<td>0.000001</td>
<td>0.000006</td>
<td>0.000006</td>
</tr>
<tr>
<td>2</td>
<td>M2</td>
<td>0.000028</td>
<td>0.000171</td>
<td>0.000099</td>
</tr>
<tr>
<td>3</td>
<td>M6</td>
<td>0.000175</td>
<td>0.000874</td>
<td>0.000408</td>
</tr>
<tr>
<td>4</td>
<td>M7</td>
<td>0.001496</td>
<td>0.005985</td>
<td>0.002617</td>
</tr>
<tr>
<td>5</td>
<td>M3</td>
<td>0.012355</td>
<td>0.037064</td>
<td>0.017254</td>
</tr>
<tr>
<td>6</td>
<td>M5</td>
<td>0.026886</td>
<td>0.053771</td>
<td>0.031296</td>
</tr>
<tr>
<td>7</td>
<td>M1</td>
<td>0.112351</td>
<td>0.112351</td>
<td>0.112351</td>
</tr>
</tbody>
</table>

6. **Conclusions**

In this paper, we are proposed a new evolutionary method for rule weights specification in fuzzy rule-based classification systems. Some best equations are presented for each dataset, were novel composite measure of confidence and support and lift for rule weighting. We are examined the performance of fuzzy if-then rules extracted from numerical data for pattern classification problems, which using new proposed weighting method. Simulation results on some well-known UCI data sets show that the new equations can lead to improve the classification performance significantly.
REFERENCES