A new hybrid approach for modeling accurate fuzzy rule based classification systems

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

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

Abstract— *we propose in this article a novel hybrid method for modelling accurate fuzzy rule based classification systems. The new method is a combination of manifold based data mapping method, a heuristic fuzzy rule based construction method and an evolutionary based rule weighting approach. Manifold based data mapping method considers the intricate geometric relationships that may exist among the data and computes a new representation of data that optimally preserves local neighbourhood information in a certain sense. Although this new representation does not secure the interpretability of obtained fuzzy models, the main intention of this research is to improve the classification accuracy significantly. To demonstrate the effectiveness of the new hybrid method in improvement classification accuracy, experiment results is reported.*

Index Terms— *Fuzzy Rule Based Classification Systems; FRBCSs; manifold learning; rule weighting; Genetic Network Programming; GNP.*

I. Introduction

In recent years, fuzzy models have been used widely because they are able to work with imprecise data, handle the complex nonlinear problems and acquired knowledge with these models is more interpretable than the black-box models. Fuzzy Rule-Based Classification System (FRBCS) is a special case of fuzzy modeling and focuses on finding a compact set of fuzzy if-then classification rules to model the input-output behavior of the system. The input of the FRBCS is a number of pre-labeled classification examples, and the output of this system is a crisp and discrete value.

Many approaches have been proposed for modeling FRBCS and learning fuzzy if-then rules from numerical data for classification problems. Below, some of them are mentioned.

Ishibuchi and Yamamoto in Ref. [1] proposed some heuristic methods for rule weight specification. Nakashima et al. 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 [2]. 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 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 [3]. Sánchez et al. combined genetic programming operators with simulated annealing to search the best rules [4]. 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 et al. proposed a novel adaboost algorithm to learn fuzzy-rule-based classifiers [5]. They applied an evolutionary boosting scheme to approximate and descriptive fuzzy-rule bases. Otero and Sánchez applied Logitboost to learn fuzzy rules in classification problems [6]. Sánchez and Otero in Ref. [7] proposed a boosting-based genetic method to learn weighted fuzzy rules. González and Pérez 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 [8]. 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 et al. combined two fuzzy genetic based machine learning approaches (i.e., Michigan and Pittsburgh) into a single hybrid algorithm [9]. Their hybrid algorithm was based on the Pittsburgh approach where a set of fuzzy rules was handled as an individual. Genetic operations for generating new fuzzy rules in the Michigan approach were utilized as a kind of heuristic mutation for partially modifying each rule set. Mansoori et al. proposed a novel steady-state genetic algorithm to extract a compact set of good fuzzy rules from numerical data (SGERD) [10]. Zolghadri and Mansoori introduced a method of fuzzy rule weight specification [11]. 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 was used as the weight of the rule. Zolghadri and Taheri employed a method of learning rule weights in fuzzy rule-based classification systems [12]. This method was 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 neighbor solution that is better than the current one. A neighbor solution is different in the value of just one parameter (the weight of one rule) compared with the current solution. Chen et al. introduced an algorithm to build an accurate classifier [13]. They proposed a framework to integrate classification and fuzzy association rule mining. Carvalho and Freitas suggested a hybrid decision tree/ genetic algorithm method to discover classification rules [14]. The central idea of this hybrid method involves the concept of small disjuncts in data mining. Gray and Fan considered the construction of classification trees using genetic algorithm [15]. This genetic algorithm has been applied to search over the space of trees by initializing a forest of randomly generated trees. Then, it has been employed for evolving the forest through the genetic operations of crossover, mutation, cloning, and transplanting to improve the performance of the trees in the forest. Gao and Wang introduced a novel center-based nearest neighbor classifier to deal with the pattern classification problems [16]. Wang et al. demonstrated that an extremely simple adaptive distance measure significantly improves the performance of the k-nearest neighbor rule [17]. Paredes and Vidal proposed a method, which minimize nearest neighbor classification error [18]. Indeed, in order to optimize the accuracy of the nearest-neighbor classification rule, a weighted distance was proposed, along with algorithms to automatically learn the corresponding weights. These weights may be specified for each class and feature or for each individual prototype or for both of them.

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 fuzzy rule-based classification system and rule weighting is employed to improve the accuracy of fuzzy rule-based classification system. The methods in the second group are better than the first ones, because adjusting weight of rules is much easier than tuning 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. The authors have recently introduced a new rule weighting method to improve the classification accuracy. For learning rule weights, an evolutionary method based on GNP [19] has been employed.

In this work, first, a geometric based algorithm is employed for mapping the data into a 2-dimensional space [20]. This algorithm uses laplacian eigenmaps for data representation and builds a graph incorporating neighborhood information of the data set, and uses the notation of laplacian of the graph for computing new representation of the data set. The representation map naturally arises from the geometry of the manifold. This algorithm attempts to preserve a different geometrical property of the underlying manifold and distance between any pairs of points in the new space is meaningful measure of distance between points [20]. Second, Data in new space are used for constructing FRBCS. For each attribute of patterns in data set, a number of pre-defined triangular fuzzy sets are used to partition the domain interval of each attribute and all fuzzy rules with two antecedent conditions are generated. Finally, the GNP based method is utilized for learning the rules' weights.

The remainder of the paper is organized as follows. General design of FRBCS from data is explained in Section II. In Section III, the algorithm for computing new representation of data is presented. The proposed method for modeling FRBCS is explained in Section IV. In Section V, experimental results are presented. Finally, we conclude the paper in Section VI.

II. General Design of Fuzzy Rule-Based Classification System

Fuzzy rule-based classification system is composed of three main 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, a simple and efficient heuristic method based on references [1, 11 and 12] is used for constructing FRBCS. Let us assume that our pattern classification problem is an n-dimensional problem with C classes and m training patterns, $X_p = [x_{p1}, x_{p2} \dots x_{pn}]$, $p = 1, 2 \dots m$. Usually, each attribute of the given training patterns is rescaled to a unit interval [0, 1] by using a linear transformation that preserves the distribution of training patterns. To partition the domain interval of each input attribute, 14 fuzzy sets showed in Fig. 1 are used. Traditionally, triangular shaped fuzzy sets are used, because they are simple and more human understandable.

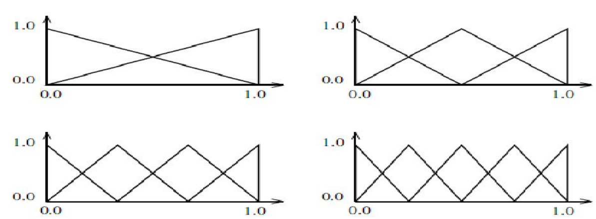

Fig. 1. Different partitioning of each attribute axis (adopted from [1]).

Fuzzy rules for a classification problem with n attributes can be written as

Rule R_q :

$$if\ x_{11}\ is\ A_{q1}\ and \dots and\ x_{1n}\ is\ A_{qn}\ then\ Class\ C_q\ with\ CF_q$$

where $X_1 = [x_{11}, x_{12} \dots x_{1n}]$ is the input attribute vector, R_q is the label of the q-th fuzzy if-then rule, $A_{q1}, A_{q2} \dots A_{qn}$ are antecedent fuzzy sets on the unit interval [0, 1], C_q is the consequent class and CF_q is the certainty grade of rule R_q.

In order to classify an input pattern $X_p = [x_{p1}, x_{p2} \dots x_{pn}]$, 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

$$\mu_q(X_p) = \prod_{i=1}^{n} \mu_{qi}(X_{pi}) \quad (1)$$

where $\mu_{qi}(.)$ is the membership function of the antecedent fuzzy set A_{qi}. 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

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

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

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

The most popular fuzzy reasoning method in FRBCSs is the reasoning based on a single winner rule. Assume the classification system has R rules, when using this method for classifying new pattern the single winner rule R_w is determined as

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

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

Fuzzy rules with two antecedent conditions are generated and product of confidence and support of rules are used as the certainty grade of rules. 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 [1].

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 compatible with the X_p or if 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 [1].

III. Using Laplacian Eigenmaps for Data Representation

This section explains the algorithm proposed by Belkin and Niyogi in [20] that is used for computing new representation of data set in *2*-dimensional space. Let us assume we have m points $x_1, ..., x_m$ in M and M is a manifold embedded in R^n (our pattern classification problem is a n-dimensional problem), and the goal is to find a set of points $y_1, ..., y_m$ in R^2 (*2*-dimensional space) such that y_i represents x_i. We construct a weighted graph with m nodes, one for each point, and a set of edges connecting neighboring points. The embedding map is now provided by computing the eigenvectors of the graph laplacian. The algorithmic procedure is formally stated below [20].

Step 1: Constructing the adjacency graph (G).

We construct a graph with m nodes, one for each point and we put an edge between node i and j if x_i and x_j are close. Nodes i and j are close and are connected by an edge if i is among k nearest neighbors of j or j is among k nearest neighbors of i ($k \in N$). Note that this relation is symmetric. The distances between points are computed by standard Euclidean distance measure.

Step 2: Calculating the edge weights (generate $m \times m$ weight matrix which denoted by W).

We use heat kernel formula for specifying the edge weights. If nodes i and j are connected, the weight of connecting edge (W_{ij}) is computed as

$$W_{ij} = e^{-\|x_i - x_j\|^2 / t}, \text{ otherwise } W_{ij} = 0 \quad (7)$$

Step 3: Eigenmaps.

Assume the graph G, constructed above, is connected, otherwise proceed with step 3 for each connected component. Compute eigenvalues and eigenvectors for the generalized eigenvector problem. $Lf = \lambda Df$, where D is diagonal weight matrix of the same size as W, its entries are column sums of W, $D_{ii} = \sum_j W_{ij}$. $L = D - W$ is the laplacian matrix. L is a Symmetric, positive semidefinite matrix which can be thought of as an operator on function defined on vertices of G. Let $f_0, ..., f_{m-1}$ be the solution of equation $Lf = \lambda Df$, ordered according to their eigenvalues (where eigenvalues denoted by λ and eigenvectors denoted by f).

$$Lf_0 = \lambda_0 Df_0,\ Lf_1 = \lambda_1 Df_1,\ ...,\ Lf_{m-1} = \lambda_{m-1} Df_{m-1}$$
$$and\ 0 = \lambda_0 \le \lambda_1 \le ... \le \lambda_{m-1} \quad (8)$$

We leave out the eigenvector f_0 corresponding to eigenvalue *0* and use the next *2* eigenvectors for embedding in *2*-dimensional Euclidean space.

$$x_i \to (f_1(i), f_2(i)) \quad (9)$$

IV. Proposed Method

Our proposed approach attempts to model an accurate FRBCS. The flowchart of the proposed algorithm is shown in Fig .2. In Part 1, the laplacian eigenmap method which explained in section III is used to compute the new representation of data and data are mapped in *2*-dimensional space. In Part 2, new representation of data are used as input data and all fuzzy rules with two antecedent variable are generated (at most $N = 14^2 = 196$ candidate rules would be generated). For determining the consequent class of each candidate rule, we use the below formula

$$Class_q = argmax\{Conf(R_q) \times Supp(R_q) \mid q = 1, \ldots, N\}. \quad (10)$$

Then, the candidate rules are divided into C groups according to their consequent classes. The candidate rules in each group are sorted in descending order of an evaluation criterion. A rule-base is constructed by choosing the best Q (Q = 196 / 2×C) fuzzy rules from each group (possibly $C.Q$ fuzzy rules in total). Among many heuristic rule evaluation measures presented in [22], we used the below evaluation measure:

$$e(R_j) = \sum_{X_t \in Class\ C_j} \mu_j(X_t) - \sum_{X_t \notin Class\ C_j} \mu_j(X_t). \quad (11)$$

After rule-base is constructed, we used an evolutionary rule weighting method based on Genetic Network Programming (GNP) that has been recently proposed by authors [19] to improve classification accuracy. This method uses some rule measures and operators and generates various combination of them (measures = {confidence, support and lift} and operators = {+, −, ×, ÷, square, square root, max, min, absolute}). Each combination will be used as an equation to specify rule weights and give classification accuracy independently. The combination that gives best accuracy is selected as the final equation used to specify rule weights. In part 3, first some GNP individuals are initialized randomly and then we calculate fitness value for all individuals; after that, evolutionary operators (i.e. selection, crossover and mutation) are applied to generate the next population. The four steps, namely fitness evaluation, selection, crossover and mutation, are executed until a termination condition is met. The maximum number of generations is considered in this study as the termination criterion. In the GNP, one individual usually could generate more than one equation thereby all these equations are evaluated and the highest accuracy is set as the fitness value of that individual. As it can be seen in part 4, for one generated equation a FRBCS is built and the equation is used to specify rule weights, then the constructed FRBCS is examined on train data and its accuracy is set as the fitness value. Finally, the equation of individual with highest fitness value is selected as the final equation that used to specify rule weights and the obtained FRBCS is examined on testing data in part 2.

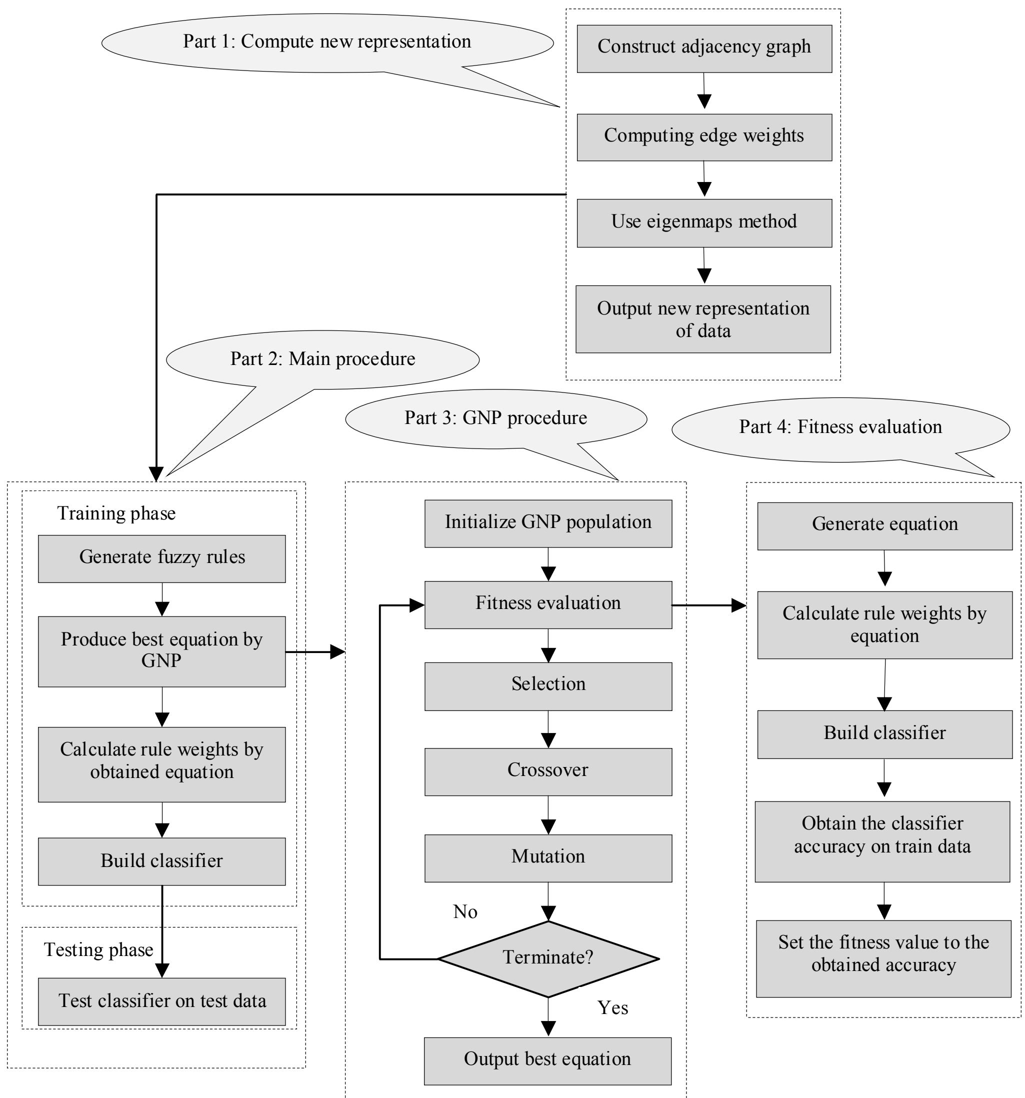

Fig. 2. The flowchart of proposed algorithm.

V. Experiment Results

A. Simulation Setup and Datasets

In this section, the performance of the new proposed method is examined. We have used 20 data sets with numerical attributes from the University of California, Irvine machine learning repository (UCI) [21], all of them valid for classification tasks. Table I shows specification of these data sets. For each data set, the name, number of examples, number of attributes and number of classes are given. The parameters setting for laplacian eigenmap method is used for computing new representations of data and the equation used to specify rule weights for each data set are shown in Table II. The parameters setting for GNP in the experiments are shown in Table III. Since the speed is one of the most important considerations of classification problems, we selected the small size GNP to limit the search space and do not spend too much time on evaluation. The previously developed FRBCS methods, which are used for performing comparisons in this study, are presented in Table IV.

TABLE I. STATISTICS OF DATASETS USED FOR PROPOSED METHOD EVALUATION.

Data set	Number of attributes	Number of examples	Number of classes
Wisconsin	9	699	2
Pima	8	768	2
Haberman	3	306	2
Bupa	6	345	2
Heart	13	270	2
Monk-2	6	432	2
Appendicitis	7	106	2
Saheart	9	462	2
Tic-tac-toe	9	958	2
Wine	13	178	3
Newthyroid	5	215	3
Iris	4	150	3
Balance	4	625	3
Post	8	90	3
Tae	5	151	3
Hayes-roth	4	160	3
Car	6	1728	4
Vehicle	18	846	4
Glass	9	214	7
Ecoli	7	336	8

TABLE II. PARAMETERS SETTING FOR LAPLACIAN EIGENMAPS IN EXPERIMENTS AND EQUATION USED TO SPECIFY RULE WEIGHTS. (*N* IS THE NUMBER OF NEAREST NEIGHBORS AND *T* IS THE PARAMETER FOR HEART KERNEL)

Data set	n	t	Equation
Wisconsin	11	2.9	(0.97×Lift)÷(0.68×min((0.98×Supp),(0.48×Conf)))
Pima	15	4	0.09×Lift
Haberman	8	1	(0.52×(0.26×Conf+0.23×Supp))−(0.76×Supp)
Bupa	19	0.6	0.56×Supp
Heart	10	6.5	square(0.28×square((0.27×Supp)−(0.11×Conf)))
Monk-2	15	5.9	min((0.28×Lift),(0.52×Supp))
Appendicitis	15	1.6	(0.5×Conf)+(0.24×Lift)
Saheart	8	4.5	(0.98×Supp)−(0.33×Conf)
Tic-tac-toe	5	5	(0.22×square(0.75×Lift))×(0.91×Conf)
Wine	7	9	(0.37×Supp)+(0.47×((0.79×Conf)−(0.18×Supp)))
Newthyroid	15	5.1	0.05×Lift
Iris	12	7.1	0.42×max((0.13×Conf),(0.3×Lift))
Balance	4	7.4	square(0.64×square(0.6×Conf)×square(0.34×Conf))
Post	5	2.1	0.31×Conf×Supp
Tae	14	5.4	(0.36×Supp)+(0.03×((0.83×Conf)÷(0.96×Supp)))
Hayes-roth	15	0.9	(0.86×Conf)÷(0.26×Supp)
Car	7	6.6	(0.81×Lift)-Supp
Vehicle	5	2.1	(0.43×Supp)−(absolute(0.3×Lift)−(0.1×Conf))
Glass	2	6.4	max((0.15×Lift),(0.47×Supp))
Ecoli	11	4.3	0.31×((0.64×Supp)−(0.18×Conf))

TABLE III. PARAMETERS SETTING FOR GNP IN THE EXPERIMENTS.

Parameter	Value
Number of individuals	5
Maximum number of generations	50
Number of judgement nodes	2

Number of processing nodes	2
Maximum transition steps	5
Rate of crossover	0.5
Rate of start node mutation	0.5
Rate of judgement node mutation	0.5
Rate of processing node mutation	0.5

TABLE IV. THE ALGORITHMS COMPARED WITH PROPOSED METHOD IN EXPERIMENTS.

Reference	Authors	Year	Method
13	Z. Chen and G. Chen	2008	M1
14	D. R. Carvalho and A. A. Freitas	2004	M2
15	J. B. Gray and G. Fan	2008	M3
7	L. Sánchez and J. Otero	2007	M4
10	E.G. Mansoori and et al	2008	M5
9	H. Ishibuchi and et al	2005	M6
1	H. Ishibuchi and T. Yamamoto	2005	M7
16	Q. Gao and Z. Wang	2007	M8
17	J. Wang and et al	2007	M9
18	R. Paredes and E. Vidal	2006	M10
2	T. Nakashima and et al	2007	M11
4	L. Sánchez and I. Couso	2001	M12
8	A. Gonzalez and R. Perez	2001	M13

We employ ten-fold cross validation (*10-CV*) testing method as a validation scheme to perform experiments and analyze results. The algorithm is run five times and the average of accuracies is calculated, 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 estimated classifier performance is the average of these 50 classification accuracies (estimated classifier accuracy is the average over 50 runs).

B. Non-parametric Statistical Tests for Comparisons

We used statistical tests to make sure that the difference is significant, that is, very unlikely to be caused by chance - the so-called *p*-value of the test [23]. To evaluate the performance of the proposed method, we are used Friedman test [23-26], 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 [25, 26]. 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, $R_j = \frac{1}{N}.\sum_i r_i^j$. 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 χ_F^2 with $k - 1$ degrees of freedom and is as follows [26]:

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

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 [25]. We use Holm's test and Finner test as post-hoc methods. Holm's test [27] 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:

$$z = \frac{(R_i - R_j)}{\sqrt{k(k+1)/6N}}. \quad (13)$$

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 α [25]. Holm's test adjusts the value for α in order to compensate for multiple comparisons.

Holm's test adjusts the value of α in a step-down manner. Let $p_1, p_2, ..., p_{k-1}$ be the ordered *p*-values (smallest to largest), so that $p_1 \le p_1 \le ... \le p_{k-1}$, $H_1, H_2, ..., H_{k-1}$ be the corresponding hypotheses. The Holm procedure rejects H_1 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 [25]. The Finner procedure [28] adjusts the value of α in a step-down manner, as Holm's method do. It rejects H_1 to H_{i-1} if i is the smallest integer so that $p_i > 1-(1-\alpha)^{(k-1)/i}$ [25].

After a post-hoc method used, the adjusted *p*-values methods are used for computing these exact *p*-values for each test procedure. The adjusted *p*-value for the Holm's procedure is computed by $p_{Holm} = (k - i)p_i$. 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* [29].

C. Results and Discussions

Experimental analysis for performance evaluation of a proposed method is a necessary task in an investigation. For the sake of comparison, Table V gives the accuracies obtained by the proposed method as well as those of previously developed methods over different data sets. In this table, we have used 20 data sets with numerical attributes from the University of California, Irvine machine learning repository (UCI) [21], all of them valid for classification tasks. We measured the performance of each classifier by means of its accuracy over 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 V, the first column shows the names of datasets. The average classification accuracy for each data set by the algorithm is proposed in this paper and the algorithms are introduced in Table IV, are showed in 2[th] to 15[th] columns, respectively. Experimental results in Table V 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.

For the sake of completeness, average ranking of the classification accuracies over different datasets (for test data) are computed and presented in Table VI. In this table, the value of Friedman statistic (distributed according to chi-square with 13 degrees of freedom) is 82.011429 and *p*-value computed by this test is 0. 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. As it can be seen, the proposed method has the best average ranking among the others. In order to sure about the significance of differences in Table VI, post hoc methods are applied over the results of this table.

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 VII. Holm's procedure rejects those hypotheses that have a *p*-value ≤ 0.016667 and Finner's procedure rejects those hypotheses that have a *p*-value ≤ 0.05. As Table VII shows, Holm's procedure verifies that the proposed method performs better than all other approaches except approaches proposed in [7] and [17], because all approaches except approaches proposed in [7] and [17] have a Holm ≤ 0.016667, and the procedure of Finner verifies that the proposed method performs better than all other approaches, because all approaches have a Finner ≤ 0.05. The adjusted *p*-values obtained are shown in Table VIII.

TABLE V. COMPARING THE CLASSIFICATION ACCURACY OF PROPOSED METHOD WITH THE OTHER CLASSIFICATION APPROACHES (10-CV TEST METHOD).

Method / Dataset	Proposed method	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13
Wisconsin	96.61	95.30	93.85	94.98	95.84	93.69	95.27	91.11	94.99	**96.86**	65.52	96.85	95.56	95.42
Pima	**78.62**	65.10	73.31	70.71	75.01	73.05	74.09	73.06	69.94	72.14	72.01	73.71	72.93	73.32
Haberman	**74.24**	51.67	73.48	71.22	73.81	73.52	72.19	73.19	68.64	68.94	73.52	73.80	71.56	71.90
Bupa	**67.85**	42.10	65.88	58.74	64.63	57.34	64.95	57.87	63.16	63.17	58.63	59.37	57.37	56.06
Heart	**80.71**	80.37	79.62	71.85	75.92	75.56	73.70	51.85	75.18	77.40	76.29	77.30	78.14	78.51
Monk-2	78.56	47.33	97.46	95.19	96.77	80.64	95.88	42.89	74.90	70.70	86.53	69.31	94.70	**97.26**
Appendicitis	**89.75**	80.18	84.09	85.81	85.00	86.90	83.00	85.81	74.45	87.63	85.72	83.72	84.09	82.18
Saheart	72.38	65.36	68.82	65.36	70.96	67.96	67.07	**72.48**	64.05	70.10	65.54	69.87	70.95	65.12
Tic-tac-toe	81.44	65.34	77.97	69.93	87.16	69.93	72.45	50.14	73.17	**81.84**	34.66	49.16	76.47	65.34
Wine	**97.80**	93.20	94.90	78.76	94.96	90.42	89.80	93.82	96.63	96.63	94.37	94.01	89.24	92.12
Newthyroid	98.00	69.80	91.64	81.77	91.70	86.53	89.35	84.24	88.44	**98.61**	97.20	97.25	92.57	90.75
Iris	**96.53**	83.33	96.00	86.00	93.33	94.00	93.33	92.66	92.66	94.66	93.33	96.00	93.33	95.33
Balance	87.73	46.08	75.49	71.34	79.34	69.12	86.39	**90.39**	74.56	89.26	76.34	85.61	74.56	74.08
Post	**77.89**	71.38	69.16	67.91	63.89	67.91	65.69	42.91	49.58	60.83	46.52	40.56	68.06	71.38
Tae	55.40	32.46	41.12	41.16	49.79	49.12	**55.83**	55.12	41.79	46.46	41.83	54.46	55.75	52.41
Hayes-roth	56.87	40.62	74.37	50.62	74.37	44.99	61.87	58.75	46.25	47.50	37.50	51.87	65.62	**77.50**
Car	**90.82**	70.02	81.36	76.26	83.21	67.18	72.22	77.83	87.67	89.69	78.02	86.22	80.26	70.02
Vehicle	53.23	32.52	71.50	47.87	48.57	52.48	50.24	60.77	**74.23**	69.39	69.26	65.20	60.50	60.00
Glass	71.91	32.89	61.16	44.53	62.77	61.02	56.60	60.04	70.94	**72.66**	70.87	58.30	55.15	58.14
Ecoli	**93.40**	42.56	76.47	62.79	70.26	74.44	73.83	72.02	67.88	82.47	82.47	80.40	70.22	84.53

TABLE VI. AVERAGE RANKINGS OF ALGORITHMS BY FRIEDMAN PROCEDURE.

Algorithm	Ranking
Proposed method	**2.7**
M1	11.75
M2	5.525
M3	10.65
M4	5.5
M5	8.975
M6	7.625
M7	8.55
M8	9.075
M9	5.15
M10	8.275
M11	6.575
M12	7.425
M13	7.225

TABLE VII. POST HOC COMPARISON TABLE FOR $\alpha = 0.05$ (FRIEDMAN).

i	Algorithm	*z*	*p*	Holm	Finner
13	M1	6.841157	0	0.003846	0.003938
12	M3	6.009635	0	0.004176	0.00786
11	M8	4.819047	0.000001	0.004545	0.011767
10	M5	4.743454	0.000002	0.005	0.015659
9	M7	4.422184	0.00001	0.005556	0.019535
8	M10	4.214308	0.000025	0.00625	0.023396
7	M6	3.72295	0.000197	0.007143	0.027242
6	M12	3.571764	0.0000355	0.008333	0.031072
5	M13	3.420578	0.000625	0.01	0.034888
4	M11	2.929225	0.003398	0.0125	0.038688
3	M2	2.135499	0.03272	0.016667	0.042474
2	M4	2.116601	0.034294	0.025	0.046244
1	M9	1.852026	0.064022	0.05	0.05

TABLE VIII. ADJUSTED P-VALUES (FRIEDMAN), PROPOSED METHOD IS THE CONTROL ALGORITHM.

i	Algorithm	unadjusted *p*	p_{Holm}	p_{Finner}
1	M1	0	0	0
2	M3	0	0	0
3	M8	0.000001	0.000016	0.000006
4	M5	0.000002	0.000021	0.000007
5	M7	0.00001	0.000088	0.000025
6	M10	0.000025	0.0002	0.000054
7	M6	0.000197	0.0001378	0.000366
8	M12	0.0000355	0.002128	0.000576
9	M13	0.000625	0.003124	0.000902
10	M11	0.003398	0.013592	0.004415
11	M2	0.03272	0.098161	0.038553
12	M4	0.034294	0.098161	0.038553
13	M9	0.064022	0.098161	0.064022

Consequently, this study proposes a novel hybrid method for generating FRBCS that is able to improve the classification accuracy significantly. The presented approach does not consider the interpretability of FRBCS because the manifold transformation maps the available data into a new space with reduced dimensions.

VI. CONCLUSIONS

We proposed in this article a new hybrid method for modeling accurate fuzzy rule based classification systems. At first a manifold based data mapping method was utilized to compute a new representation of data set that arises from the geometry of the manifold. The manifold assumptions caused to better classification rates and helped to increase the speed of the classification process (this is particularly relevant in high dimensional problems). This new representation map of data set was used as input data for generating FRBCS based on a heuristic approach. Then rule weighting used as a simple mechanism to improve classification performance. We employed an evolutionary method based on genetic network programming (that has been recently introduced by authors) for rule weights specification. Some equations were obtained via this method for each data set that were novel composite measures of confidence, support and lift for rule weighting. In order to illustrate the efficiency of the new method, several experiments were performed over multiple data sets taken from UCI repository. Also for comparing the proposed method against other well known previously developed methods, several statistical tests were done. Simulation results show that the new approach significantly improves classification performance.

REFERENCES

[1] H. Ishibuchi, T. Yamamoto, Rule Weight Specification in Fuzzy Rule-Based Classification Systems, IEEE Trans. on Fuzzy Systems, (2005), Vol. 13, No. 4, pp. 428-435.

[2] T. Nakashima, G. Schaefer, Y. Yokota, H. Ishibuchi, A weighted fuzzy classifier and its application to image processing tasks, Fuzzy Sets and Systems, (2007), Vol. 158, No. 3, pp. 284 – 294.

[3] Y. Chen, J. Z. Wang, Support Vector Learning for Fuzzy Rule-Based Classification Systems, IEEE Trans. on Fuzzy Systems (2003), Vol. 11, No. 6, pp. 716-728.

[4] L. Sánchez, I. Couso, J. A. Corrales, Combining GP operators with SA search to evolve fuzzy rule based classifiers, Information Sciences, (2001), Vol. 136, pp. 175-191.

[5] M. J. Jesus, F. Hoffmann, L. J. Navascués, L. Sánchez, Induction of Fuzzy-Rule-Based Classifiers With Evolutionary Boosting Algorithms, IEEE Trans. On Fuzzy Systems (2004), Vol. 12, No. 3, pp. 296-308.

[6] J. Otero, L. Sánchez, Induction of descriptive fuzzy classifiers with the Logitboost algorithm, Soft Computing, (2006), Vol. 10, No. 9, pp. 825-835.

[7] L. Sánchez, J. Otero, Boosting Fuzzy Rules in Classification Problems Under Single-Winner Inference, International journal of intelligent systems, (2007), Vol. 22, No. 9, pp. 1021-1034.

[8] A. González, R. Pérez, Selection of Relevant Features in a Fuzzy Genetic Learning Algorithm, IEEE Trans. on Systems, Man, and cybernetics—part b: cybernetics, (2001), Vol. 31, No. 3, pp. 417-425.

[9] H. Ishibuchi, T. Yamamoto, T. Nakashima, Hybridization of fuzzy GBML approaches for pattern classification Problems, IEEE Trans. on Systems, Man, and Cybernetics- Part B: Cybernetics, (2005), Vol. 35, No. 2, pp. 359- 365.

[10] E. G. Mansoori, M. J. Zolghadri, and S. D. Katebi, SGERD: A Steady-State Genetic Algorithm for Extracting Fuzzy Classification Rules From Data, IEEE Trans. on fuzzy systems, (2008), Vol. 16, No. 4, pp. 1061-1071.

[11] M. J. Zolghadri, E. G. Mansoori, Weighting fuzzy classification rules using receiver operating characteristic (ROC) analysis, Information Sciences, (2007), Vo. 177, No. 11, pp. 2296-2307.

[12] M. J. Zolghadri, M. Taheri, A proposed method for learning rule weights in fuzzy rule-based classification systems, Fuzzy Sets and Systems, (2008), Vol. 159, No. 4, pp. 449-459.

[13] Z. Chen, G. Chen, Building an associative classifier based on fuzzy association rules, International Journal of Computational Intelligence systems, (2008) Vol. 1, No. 3, pp. 262-273.

[14] D. R. Carvalho and A. A. Freitas, A hybrid decision tree/genetic algorithm method for data mining, Information Sciences, (2004), Vo. 163, pp. 13-35.

[15] J. B. Gray and G. Fan, Classification tree analysis using TARGET, Computational Statistics & Data Analysis, (2008), Vol. 52, No. 3, pp. 1362-1372.

[16] Q . Gao,Z. Wang, Center-based nearest neighbor classifier, Pattern Recognition, (2007), Vol. 40, No. 1, pp. 346-349.

[17] J. Wang, P. Neskovic, L. N. Cooper, Improving nearest neighbor rule with a simple adaptative distance measure, Pattern Recognition Letters, (2007), Vol. 28, No. 2, pp. 207-213.

[18] R. Paredes, E. Vidal, Learning weighted metrics to minimize nearest-neighbor classification error, IEEE Trans. on Pattern Analysis and Machine Intelligence, (2006), Vol. 28, No. 7, pp. 1100-1110.

[19] F. Farahbod, M. Eftekhari, An Evolutionary Approach For Learning Rule Weights In Fuzzy Rule-Based Classification Systems, International Journal of Fuzzy Logic Systems(IJFLS), Vol. 2, No. 3 July 2012, pp. 1-15.

[20] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation, (2003), Vol. 15, No. 6, pp. 1373–1396.

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

[22] H. Ishibuchi, T. Yamamoto, Comparison of heuristic criteria for fuzzy rule selection in classification problems, Fuzzy Optimization and Decision Making, (2004), Vol. 3, No. 2, pp. 119-139.

[23] O. T. Yílíz, ö. Aslan, E. Alpaydin, Multivariate Statistical Tests for Comparing Classification Algorithms, 5th international conference on learning and intelligent optimization, Springer-Verlag Berlin, Heidelberg, pp. 1–15, (2011).

[24] D. Sheskin, Handbook of parametric and nonparametric statistical procedures. Chapman & Hall/CRC, (2006).

[25] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Information Sciences, (2010), Vol. 180, No. 10, pp. 2044–2064.

[26] J. Demšar, Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research, (2006), Vol. 7, pp. 1-30.

[27] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian Journal of Statistics (1979), Vol. 6, No. 2, pp. 65–70.

[28] H. Finner, On a monotonicity problem in step-down multiple test procedures, Journal of the American Statistical Association, (1993), Vol. 88, No. 423, pp. 920–923.

[29] J. Alcalá-fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework, J. of Mult.-Valued Logic & Soft Computing (2011), Vol. 17, pp. 255–287.