Journal of Computing and Security

April 2014, Volume 1, Number 2 (pp. 111–122)

http://www.jcomsec.org

A New Hybrid Approach for Modeling Accurate Fuzzy Rule Based Classification Systems

Fahimeh Farahbod [a,*]
Mahdi Eftekhari [a]

[a] *Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman, Iran.*

ARTICLE INFO.

Article history:
Received: 04 January 2013
Revised: 23 June 2013
Accepted: 29 November 2013
Published Online: 13 September 2014

Keywords:
Fuzzy Rule Based Classification Systems, FRBCSs, Manifold Learning, Rule Weighting, Genetic Network Programming, GNP.

ABSTRACT

we propose in this article a new hybrid method for modeling 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 neighborhood information in a certain sense. Although this new representation does not secure the interpret ability of obtained fuzzy models, the main intention of this research is to improve the classification accuracy significantly. Experiments on some well-known datasets are performed to show the performance of the new proposed approach. Some nonparametric statistical tests are used to analysis the results obtained in experiments. Experimental results confirm the effectiveness of our proposed method in improvement of the classification accuracy.

1 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 consider a linguistic fuzzy rule structure where the output involves a discrete value, the class associated to the patterns matching the rule antecedent.

Many approaches have been proposed for learning rules for classification problems. Below, some of them are mentioned.

Berzal and et al. proposed a method for modeling a classification system by Rule induction. This method induces a set of rules based on ideas from the association rule mining context. It builds decision lists that can be viewed as degenerate, polythetic decision trees, by means of a Separate and Conquer strategy [1]. Kurgan and et al. proposed a simple algorithm for rule induction. This algorithm composed by two main phases. In the first one, a data reduction technique is applied in order to condensate both the positive and negative examples of each rule. The second stage uses a hill climbing methods to build effective rules that covers the maximum number of positive examples with no cover of negative examples. Two threshold can be used for alleviate specialization of rules, pruning or generalizing them [2].

Pham and Afify proposed an algorithm that employs a fast a noise-tolerant search method for ex-

* Corresponding author.
Email addresses: farahbod.ai@gmail.com (F. Farahbod), m.eftekhari@uk.ac.ir (M. Eftekhari)
ISSN: 2322-4460

tracting IF-THEN rules from examples. The method generates the best candidate rules for each example of the training data. The best is that one which covers the seed example and as many positive examples and as less negative examples as possible [3]. Bacardit and Garrell proposed a method to extract a set of maximally accurate rules that completely defines the feature space. This method models a Pittsburgh-style learning classifier system. The core of the system consists of a Genetic Algorithm which evolves individuals formed by a set of production rules. The individuals are evaluated according to the proportion of correct classified training examples [4]. Genetic algorithms for rule discovery can be divided into two broad approaches, based on how rules are encoded in the population of individuals (chromosomes). In the Michigan approach each individual encodes a single prediction rule, where in the Pittsburgh approach each individual encodes a set of prediction rules.

Aguilar-Ruiz and et al. proposed a method to extract a set of hierarchical decision rules for classification in continuous and discrete domains. This method produces a hierarchical set of rules, that is, the rules are sequentially obtained and must be, therefore, tried in order until one is found whose conditions are satisfied. In order to extract the rule-list a real-coded Genetic Algorithm (GA) is employed in the search process. Two genes will define the lower and upper bounds of the rule attribute. One rule is extracted in each iteration of the GA and all the examples covered by that rule are deleted. A parameter epf (examples pruning factor) defines a percentage of examples that can remain uncovered. Thus, the termination criterion is reached when there are no more examples to cover, depending on the epf [5].

Sousa and et al. proposed a method for determining a set of rules that predicts correctly the value of the target attribute. In this work classifier is built by a hybridization of Particle Swarm Optimization and Ant Colony Optimization [6]. Parpinelli and et al. proposed a new algorithm for rule learning. This algorithm is based in an ant colony system. In this case, the rule stands for the path that the ant must follow. Each ant starts with an empty rule and the decision of adding a new term depends on a heuristic function and a pheromone value. The heuristic function is the entropy measure for each attribute-value. There is also a prune step that removes one by one a term of the rule while this process improves the quality of that rule. Once the antecedent of the rule is totally built, the system chooses the as consequent class the majority class of the covered examples. The algorithm then selects the best ant/rule of the current iteration and adds it to the rule-set. This process iterates until all examples are covered (depending on the parameters of the user) [7].

Bacardit and et al. proposed a method to perform a classification task by means of a GA-based algorithm. This method is a GA algorithm whose uses the Michigan approach [8]. Tan and et al. proposed a co-evolutionary algorithm for rules discovery. In this approach, each individual codifies a rule, and the whole rule set is evolved simultaneously. Thus, rules should cooperate among them to get an optimal rule set jointly, and at the same time, rules compete against each other to survive in the population. This algorithm evolves a set of rules, which are initialized randomly, using as fitness a combination of the true positive rate and the false positive rate, together with a token competition that reduces the size of the rule-set. It uses a specific regeneration operator that re initializes those chromosomes that have fitness below the average. For nominal attributes it uses the one-point crossover, whereas for the numerical attributes it applies a linear combination of the parents [9].

Jiao and et al. proposed a method for modeling a genetics-based rule based system. This method creates groups of similar examples with the same class label and then builds a rule for each group discovered. Thus the chromosome has a variable length and it stores the position of the example in the training set. At classification time, the most general rule that covers the example defines its class [10]. Ishibuchi and Yamamoto proposed some heuristic methods for rule weight specification [11]. 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. 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 [12]. Sanchez and Otero proposed a boosting-based genetic method to learn weighted fuzzy rules [13].

Gonzalez and Perez 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 [14]. 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. 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 [15].

Mansoori et al. proposed a novel steady-state genetic algorithm to extract a compact set of good fuzzy rules from numerical data (SGERD) [16]. Sanchez et al. 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 macro mutation operator from tree-shaped genotype genetic algorithms as adjacency operator [17]. Luaces and Bahamonde proposed an algorithm that combines rule induction and instance-based learning. The algorithm tries to extract a small set of suitable rules to represent suitable the training set, achieving an acceptable accuracy. This algorithm first selects a small set of instances, and converts them to punctual rules. They are iteratively inflated in order to cover as many training examples as possible. This process is repeated several times. Then, the complete set of rules extracted is post processed, to improve its classification performance and remove redundant rules [18].

Chen et al. introduced an algorithm to build an accurate classifier. They proposed a framework to integrate classification and fuzzy association rule mining [19]. Carvalho and Freitas suggested a hybrid decision tree/ genetic algorithm method to discover classification rules. The central idea of this hybrid method involves the concept of small disjuncts in data mining [20]. Gray and Fan considered the construction of classification trees using genetic algorithm. 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 [21]. Gao and Wang proposed a center-based nearest neighbor classifier [22]. Wang et al. proposed a method for improving nearest neighbor rule [23]. Paredes and Vidal proposed a method for learning weighted metrics to minimize nearest-neighbor classification error [24].

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.

In this study, rule weight is used as a simple mechanism to improve the classification accuracy, 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 Genetic Network Programming (GNP)[25] has been employed.

In this work, first, a geometric based algorithm is employed for mapping the data into a 2-dimensional space [26]. Usually, algorithms that are used for classification problems don't consider the intricate geometric relationships that may exist among the data. We use a data mapping algorithm, which considers the intricate geometric relationships that may exist among the data and computes a new representation of data that optimally preserves local neighborhood information in a certain sense. 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 [26]. 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, an evolutionary based rule weighting method [25] is utilized for further improvements of the classification accuracy.

The remainder of the paper is organized as follows. General design of FRBCS from data is explained in Section 2. In Section 3, the algorithm for computing new representation of data is presented. The proposed method for modeling FRBCS is explained in Section 4. In Section 5, experimental results are presented. Finally, we conclude the paper in Section 6.

2 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 describes the semantic of fuzzy sets associated to linguistic labels. 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". The reasoning method provides the mechanism to classify a pattern using the information from the rule-base and database.

In this work, a simple and efficient heuristic method based on reference [11] is used for constructing FRBCS. 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 Figure 1 are used. Traditionally, triangular shaped fuzzy sets are used, because they are simple and more human understandable.

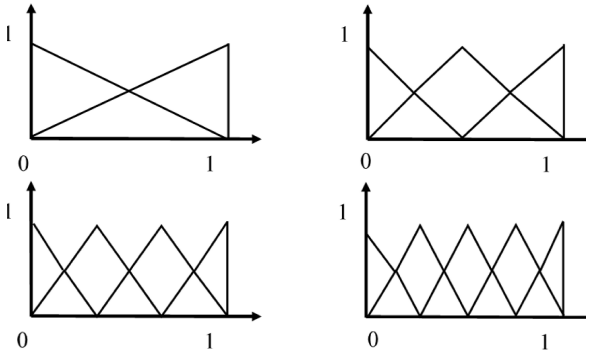

Figure 1. Different partitioning of each attribute axis (adopted from [11]).

Let us assume that our pattern classification problem is an n dimensional problem with C classes and m training patterns, $X_p = [x_{p1}, x_{p2}, \ldots, x_{pn}], p = 1, 2 \ldots m$. Fuzzy rules for a classification problem with n attributes can be written as

Rule R_q is

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

where $X_1 = [x_{11}, x_{12} \ldots x_{1n}]$ 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 and CF_q is the certainty grade of rule R_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 rule R_q with the input pattern X_p can be calculated as

$$\mu_q(x_p) = \prod_{i=1}^{n} \mu_{qi}(x_{pi}) \tag{2}$$

where $\mu_q i(\dot{})$ 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

$$Conf(A_q \rightarrow ClassC_q) = \sum_{x_p \in ClassC_q} \frac{\mu_q(X_p)}{\sum_{p=1}^{m} \mu_q(X_p)} \tag{3}$$

$$Supp(A_q \rightarrow ClassC_q) = \frac{1}{m} \sum_{x_p \in ClassC_q} \mu_q(X_p) \tag{4}$$

$$Lift(A_q \rightarrow ClassC_q) = \frac{\sum_{x_p \in ClassC_q} \frac{\mu_q(X_p)}{\sum_{p=1}^{m} \mu_q(X_p)}}{\frac{r}{m}} \tag{5}$$

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

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

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

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 [11].

The new pattern X_p is classified as class C_w, which is the consequent class of the winner rule R_w. If multiple fuzzy rules have the same maximum value but different consequent classes for the new pattern X_p in Equation 6, the classification of X_p is rejected. The classification is also rejected if no fuzzy rule is compatible with the new pattern X_p [11].

3 Using laplacian eigenmaps for data representation

This section explains the algorithm proposed by Belkin and Niyogi in [26] that is used for computing new representation of data set in 2-dimensional space. Let us assume we have m points $x_1, \ldots, x_m$ in M and M is a manifold embedded in R^n (our pattern classification problem is an n-dimensional problem), and the goal is to find a set of points $y_1, \ldots, y_m$ 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

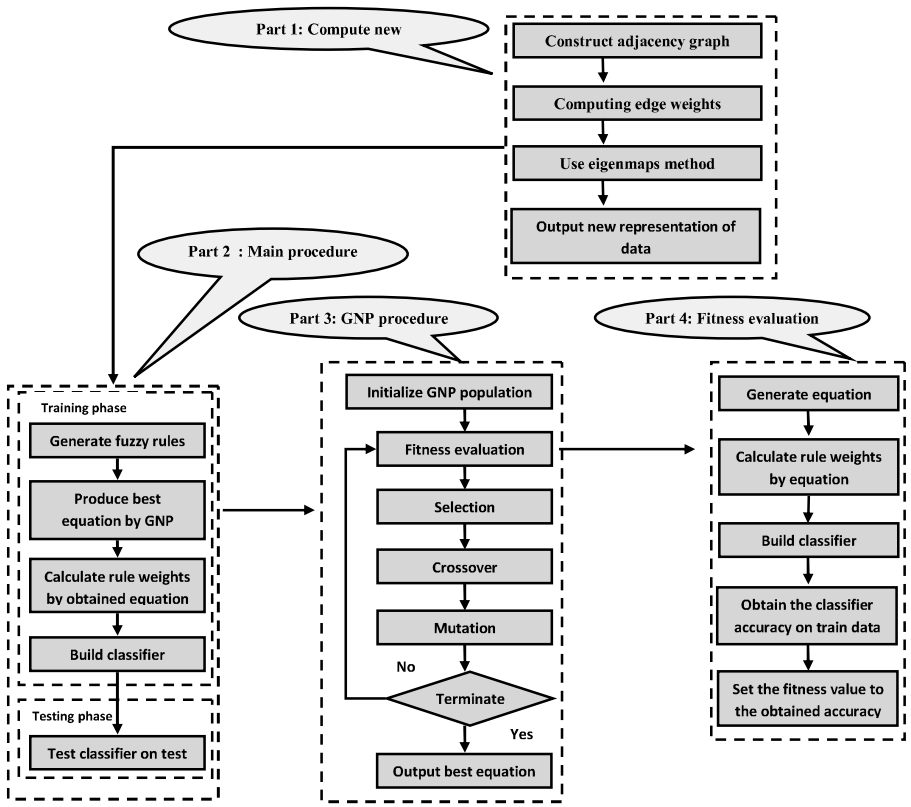

Figure 2. The flowchart of proposed algorithm.

procedure is formally stated below [26].

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 We generate $m \times m$ weight matrix which denoted by W and 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

$$\begin{aligned} w_{ij} &= e^{\frac{-\|x_i - x_j\|^2}{t}}, \\ otherwise \quad w_{ij} &= 0 \end{aligned} \tag{7}$$

Step 3: Eigenmaps We compute Eigenvalues and Eigenvectors for the generalized Eigenvector problem $L.f = \lambda.D.f$. The entries of the matrix D are column sums of W, $D_{ii} = \sum_j w_{ij}$.$L = D - W$ is the laplacian matrix which can be thought of as an operator on function defined on vertices's of G. Let $f_0, \ldots, f_{m-1}$ be the solution of equation $L.f = \lambda.D.f$, ordered according to their eigenvalues (where eigenvalues denoted by λ and eigenvectors denoted by f).

$$Lf_0 = \lambda_0 D f_0, \ldots, Lf_{m-1} = \lambda_{m-1} D f_{m-1} \tag{8}$$

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

$$x_i \to (f_1(i), f_2(i)) \tag{9}$$

4 Proposed method

Our proposed approach attempts to model an accurate FRBCS. The flowchart of the proposed algorithm is shown in Figure 2 Lets explain the four parts in Figure 2 with more details.

In Part 1, the laplacian eigenmap method which explained in Section 3 is used to compute the new representation of data and data are mapped in 2-dimensional space. Using this new representation is led to better classification rates and helped to increase the speed of the classification process (this is particularly relevant in high dimensional problems).

Part 2 consists of the most general steps of proposed method. At first, new representation of data are used as input data and all fuzzy rules with two antecedent variable are generated (at most $N = 14^2 = 196$ candi-

date rules would be generated). For determining the consequent class of each candidate rule, we use the below formula

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

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 $Q(Q = 196/(2\times C))$ fuzzy rules from each group (possibly $C\times Q$ fuzzy rules in total). Among many heuristic rule evaluation measures presented in [27], we used the below evaluation measure:

$$e(R_j) = \sum_{x_t \in ClassC_j} \mu_j(X_t) - \sum_{x_t \notin ClassC_j} \mu_j(X_t) \quad (11)$$

After the fuzzy rules have been generated, we use an evolutionary rule weighting method that has been recently proposed by authors [25] to get some rule weight equations, and use the best equation to specify rule weights. This method uses some rule measures and operators and generates various combination of them ($measures = confidence,\ support\ and\ lift$ and $operators = +, -, \times, \div, square, squareroot, max, min, absolute$). 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. As soon as the rule weights have been specified, classifier building procedure will be carried out.

Part 3 shows the steps of rule weighting method. In this part, 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.

In part 4, we could see how to evaluate each rule weight equation. After one equation has been generated, an 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.The data for testing here is the training data (not the testing data), since all the steps of part 4 are within the training procedure of part 2.

Table 1. Statistics of datasets used for proposed method evaluation.

Dataset	No. of Attributes	No. of Instances	No. 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

5 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) [28], all of them valid for classification tasks. Table 1 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

Table 2. 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)

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

new representations of data and the equation used to specify rule weights for each data set are shown in Table 2. The parameters setting for GNP in the experiments are shown in Table 3. 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 methods, which are used for performing comparisons in this study, are presented in Table 4.

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

Table 3. 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 4. The algorithms compared with proposed method in experiments.

Reference	Authors	Year	Method
19	Z. Chen and G. Chen	2008	M1
20	D. R. Carvalho and A. A. Freitas	2004	M2
21	J. B. Gray and G. Fan	2008	M3
13	L. Sanchez and J. Otero	2007	M4
16	E.G. Mansoori and et al	2008	M5
15	H. Ishibuchi and et al	2005	M6
11	H. Ishibuchi and T. Yamamoto	2005	M7
22	Q. Gao and Z. Wang	2007	M8
23	J. Wang and et al	2007	M9
24	R. Paredes and E. Vidal	2006	M10
12	T. Nakashima and et al	2007	M11
17	L. Sanchez and I. Couso	2001	M12
14	A. Gonzalez and R. Perez	2001	M13

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 use 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 [29]. To evaluate the performance of the proposed method, we use Friedman test [29], [30], 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 [31] , [32]. Friedman's tests 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 [32]:

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

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 [31]. We use Holms test and Finner test as post-hoc methods. Holms test [33] 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:

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

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 [31]. Holms test adjusts the value for α in order to compensate for multiple comparisons.

Holms test adjusts the value of in a step-down

Table 5. Comparing the classification accuracy of proposed method with the other classification approaches(10-CV test method).

Dataset	PM	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

manner. Let $p_1, p_2, \ldots, p_{k-1}$ be the ordered p-values (smallest to largest), so that $p_1 \leq p_2 \leq \cdots \leq p_{k-1}$, $H_1, H_2, \ldots, H_{k-1}$ be the corresponding hypotheses. The Holm procedure rejects H_1 to H_{i-1} if i is the smallest integer such that $p_i > \frac{\alpha}{(k-i)}$. Holms step-down procedure starts with the most significant p-value. If p_1 is below $\frac{\alpha}{(k-1)}$, the corresponding hypothesis is rejected and we are allowed to compare p_2 with $\frac{\alpha}{(k-2)}$. 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 [31]. The Finner procedure [34] adjusts the value of α in a step-down manner, as Holms method do. It rejects H_1 to H_{i-1} if i is the smallest integer so that $p_i > 1 - (1-\alpha)^{\frac{(k-1)}{i}}$ [31].

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 [35].

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 5 gives the accuracies obtained by the proposed method as well as those of previously developed methods over different data sets. 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 5, 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 6, are showed in 2^th to 15^th columns, respectively. Experimental results in Table 5 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 6. 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 6, 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 7. Holm's procedure rejects those hypotheses that have a $p - value \leq 0.016667$ and Finner's procedure rejects those hypotheses that have a $p - value \leq 0.05$. As Table 7 shows, Holm's procedure verifies that the proposed method performs better than all other approaches except approaches proposed in [13] and [23], because all approaches except approaches proposed in [13] and [23] have a $Holm \leq 0.016667$, and the procedure of Finner verifies that the proposed method performs better than all other approaches, because all approaches have a $Finner \leq 0.05$. The adjusted p-values obtained are shown in Table 8.

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 interpret ability of FRBCS because the manifold transformation maps the available data into a new space with reduced dimensions.

Table 6. 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 7. 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

6 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

Table 8. 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

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] Fernando Berzal, Juan-Carlos Cubero, Daniel Sanchez, and Jose Maria Serrano. Art: A hybrid classification model. *Machine Learning*, 54(1): 67–92, 2004.

[2] Lukasz A. Kurgan, Krzysztof J. Cios, and Scott Dick. Highly scalable and robust rule learner: Performance evaluation and comparison. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 36(1):32–53, 2006.

[3] Duc Truong Pham and A. A. Afify. Rules-6: A simple rule induction algorithm for supporting decision making. *31st Annual Conference of IEEE Industrial Electronics Society*, pages 2184–2189, 2005.

[4] Jaume Bacardit and Josep Maria Garrell. Bloat control and generalization pressure using the minimum description length principle for a pittsburgh approach learning classifier system. *International conference on Learning classifier systems, Lecture Notes in Computer Science Volume*, 4399:59–79, 2007.

[5] Jesus S. Aguilar-Ruiz, Raul Giraldez, and Jose Cristobal Riquelme Santos. Natural encoding for evolutionary supervised learning. *IEEE Transactions on Evolutionary Computation*, 11(4), 2007.

[6] Tiago Sousa, Arlindo Silva, and Ana Neves. Particle swarm based data mining algorithms for classification tasks. *Parallel Computing*, 30:767–783, 2004.

[7] Rafael S. Parpinelli, Heitor S. Lopes, and Alex A. Freitas. Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 6(4):321–332, 2002.

[8] Jaume Bacardit, Edmund K. Burke, and Natalio Krasnogor. Improving the scalability of rule-based evolutionary learning. *Memetic Computing*, 1(1):55–67, 2009.

[9] K. C. Tan, Q. Yu, and J. H. Ang. A coevolutionary algorithm for rules discovery in data mining. *International Journal of Systems Science*, 37(12):835–864, 2006.

[10] Licheng Jiao, Jing Liu, and Weicai Zhong. An organizational co-evolutionary algorithm for classification. *IEEE Transactions on Evolutionary Computation*, 10(1):67–80, 2006.

[11] Hisao Ishibuchi and Takashi Yamamoto. Rule weight specification in fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 13(4):428–435, 2005.

[12] Tomoharu Nakashima, Gerald Schaefer, Yasuyuki Yokota, and Hisao Ishibuchi. A weighted fuzzy classifier and its application to image processing tasks. *Fuzzy Sets and Systems*, 158(3):284–294, 2007.

[13] Luciano Sanchez and Jose Otero. Boosting fuzzy rules in classification problems under single-winner inference. *International journal of intelligent systems*, 22(9):1021–1034, 2007.

[14] Antonio Gonzalez and Raul Perez. Selection of relevant features in a fuzzy genetic learning algorithm. *IEEE Transactions on Systems, Man, and cyberneticspart b: cybernetics*, 31(3):417–425, 2001.

[15] Hisao Ishibuchi, Takashi Yamamoto, and To-

moharu Nakashima. Hybridization of fuzzy gbml approaches for pattern classification problems. *IEEE Transactions on Systems, Man, and Cybernetics- Part B: Cybernetics*, 35(2):359–365, 2005.

[16] Eghbal. G. Mansoori, Mansoor. J. Zolghadri, and Seraj. D. Katebi. Sgerd: A steady-state genetic algorithm for extracting fuzzy classification rules from data. *IEEE Transactions on fuzzy systems*, 16(4):1061–1071, 2008.

[17] Luciano Sanchez, Ines Couso, and J. A. Corrales. Combining gp operators with sa search to evolve fuzzy rule based classifiers. *Information Sciences*, 136:175–191, 2001.

[18] Oscar Luaces and Antonio Bahamonde. Inflating examples to obtain rules. *International Journal of Intelligent Systems*, 18:1113–1143, 2003.

[19] Zuoliang Chen and Guoging Chen. Building an associative classifier based on fuzzy association rules. *International Journal of Computational Intelligence systems*, 1(3):262–273, 2008.

[20] Deborah R. Carvalho and Alex A. Freitas. A hybrid decision tree/genetic algorithm method for data mining. *Information Sciences*, 163:13–35, 2004.

[21] J. Brian Gray and Guangzhe Fan. Classification tree analysis using target. *Computational Statistics and Data Analysis*, 52(3):1362–1372, 2008.

[22] Qing-Bin Gao and Zheng-Zhi Wang. Center-based nearest neighbor classifier. *Pattern Recognition*, 40(1):346–349, 2007.

[23] Jigang Wang, Predrag Neskovic, and Leon N. Cooper. Improving nearest neighbor rule with a simple adaptive distance measure. *Pattern Recognition Letters*, 28(2):207–213, 2007.

[24] Roberto Paredes and Enrique Vidal. Learning weighted metrics to minimize nearest-neighbor classification error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7): 1100–1110, 2006.

[25] Fahimeh Farahbod and Mahdi Eftekhari. An evolutionary approach for learning rule weights in fuzzy rule-based classification systems. *International Journal of Fuzzy Logic Systems(IJFLS)*, 2 (3):1–15, 2012.

[26] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[27] A. Frank and A. Asuncion. http://archive.ics.uci.edu/ml. *UCI Machine Learning Repository*, 2010.

[28] Hisao Ishibuchi and Takashi Yamamoto. Comparison of heuristic criteria for fuzzy rule selection in classification problems. *Fuzzy Optimization and Decision Making*, 3(2):119–139, 2004.

[29] Olcay Taner Yiliz, Ozlem Aslan, and Ethem Alpaydin. Multivariate statistical tests for comparing classification algorithms. In *5th international conference on learning and intelligent optimization, Springer-Verlag Berlin, Heidelberg*, 2011.

[30] David Sheskin. Handbook of parametric and nonparametric statistical procedures. *Chapman and Hall/CRC*, 2006.

[31] Salvador Garcia, Alberto Fernandez, Julian Luengo, and Francisco Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044–2064, 2010.

[32] Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[33] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[34] H. Finner. On a monotonicity problem in step-down multiple test procedures. *Journal of the American Statistical Association*, 88(423):920–923, 1993.

[35] J. Alcala-fdez, A. Fernandez, J. Luengo, J. Derrac, S. Garcia, L. Sanchez, and F. Herrera. Keel data-mining software tool:data set repository,integration of algorithms and experimental analysis framework. *Journal of Multi-Valued Logic and Soft Computing*, 17:255–287, 2011.

Fahimeh Farahbod received her bachelor's degree in hardware engineering from Department of Computer Science and Engineering Shiraz University, Shiraz, Iran, in 2008. She also received her master's degree in Artificial Intelligence from Shahid Bahonar University of Kerman, Kerman, Iran in 2012. Her research interests include Fuzzy Systems, Data Mining, Information Extraction, Machine Learning, Heuristic Search, and Database Security.

Mahdi Eftekhari was born in Kerman, Iran in 1978. He received his B.Sc. in computer engineering from Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran in September 2001. He obtained his M.Sc. and Ph.D. degrees in Artificial Intelligence from the same department in 2004 and 2008 respectively. He has been a faculty member of Computer Engineering Department at Shahid Bahonar University of Kerman, Kerman, Iran since 2008. His research interests include Fuzzy systems and modeling, Evolutionary Algorithms, Data Mining, Machine Learning and Application of intelligent methods in bioinformatics. He is the author and co-author of about 70 papers in cited journals and conferences. Dr. Eftekhari is a member of Iranian Society of Fuzzy Systems.