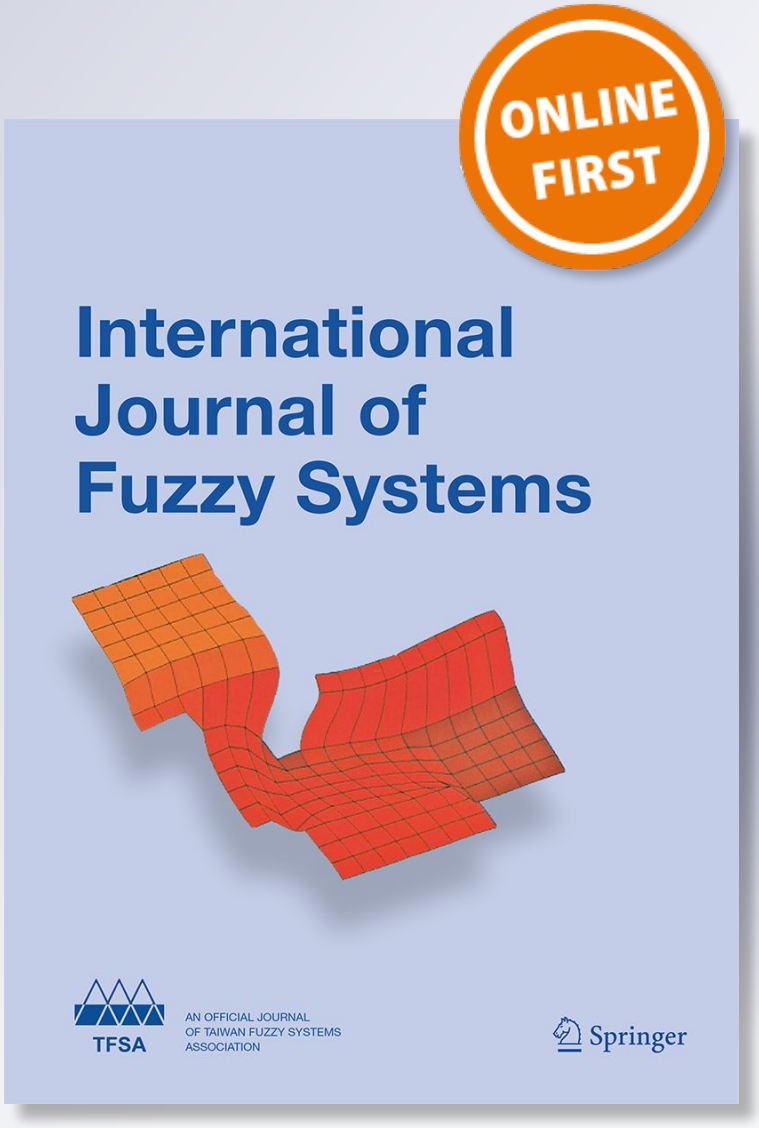
Surrogate-Assisted Multi-objective Genetic Algorithms for Fuzzy Rule-Based Classification

Harihar Kalia, Satchidananda Dehuri, Ashish Ghosh & Sung-Bae Cho

International Journal of Fuzzy Systems

ISSN 1562-2479

Int. J. Fuzzy Syst.
DOI 10.1007/s40815-018-0478-3



Springer

Int. J. Fuzzy Syst.
https://doi.org/10.1007/s40815-018-0478-3

Surrogate-Assisted Multi-objective Genetic Algorithms for Fuzzy Rule-Based Classification

Harihar Kalia[1] · Satchidananda Dehuri[2] · Ashish Ghosh[3] · Sung-Bae Cho[4]

Received: 25 May 2017 / Revised: 7 November 2017 / Accepted: 29 December 2017

Abstract In this paper, we present a surrogate-assisted multi-objective genetic algorithm to mine a small number of linguistically interpretable fuzzy rules for high-dimensional classification task in the realm of data mining. However, the difficulties like (1) handling of high-dimensional problems by fuzzy rule-based systems (i.e., the exponential increase in the number of fuzzy rules with the number of input variables), (2) the deterioration in the comprehensibility of fuzzy rules when they involve many antecedent conditions, and (3) the optimization of multiple objectives in fuzzy rule-based system may stand as pertinent issues. Hence, to combat with the aforesaid issues, we design the problem as a combinatorial optimization problem with three objectives: to maximize the number of correctly classified training patterns, minimize the number of fuzzy rules, and minimize the total number of antecedent conditions. We optimize these objectives of the fuzzy rule-based system by using a multi-objective genetic algorithm. Further to reduce the number of required evaluations to approximate the Pareto front of computationally expensive multi-objective genetic algorithm for fuzzy rule-based system, a radial basis neural network surrogate model is adapted. This approach searches for non-dominated rule sets with respect to these three objectives. The performance of the surrogate-assisted model is evaluated through a few benchmark datasets obtained from knowledge extraction based on evolutionary learning data repository. The experimental outcome confirm that this model is competitive compared to the non-surrogate-assisted model. However, the performance of the model has drawn a clear edge over rule mining approaches like Decision Table, JRip, OneR, PART, and ZeroR.

Keywords Classification rule · Fuzzy set · Genetic algorithm · Multi-objective genetic algorithm · Surrogate-assisted model

1 Introduction

Data mining [1] is an iterative process used as a central step in knowledge discovery process, performed through either semiautomatic, or automatic methods. In particular, it involves designing algorithms for tasks like classification, association rule mining, associative classification, clustering, data visualization, and sequential pattern analysis for exploring the data, developing model, and discovering previously unknown patterns. Machine learning, on the otherhand involves the study of construction of efficient and accurate algorithms from empirical data/examples is a major research area. Hence, the synergy of data mining and machine learning can be realized as a greater force for solving the tasks of data mining effectively. In many real-

✉ Sung-Bae Cho
sbcho@yonsei.ac.kr

Harihar Kalia
smarahari07@gmail.com

[1] Department of Computer Science, Seemanta Engineering College, Mayurbhanj, Odisha 757086, India

[2] Department of Information and Communication Technology, Fakir Mohan University, Balasore, Odisha 756019, India

[3] Machine Intelligence Unit and Center for Soft Computing Research, Indian Statistical Institute, 203 B.T. Road, Kolkata 700108, India

[4] Department of Computer Science, Yonsei University, Soft Computing Laboratory, 134 Shinchon-dong, Sudaemoon-gu, Seoul 120-749, South Korea

Published online: 22 March 2018

life situations, the decision-making problem is treated as a classification problem [2] of data mining.

Classification is a supervised learning technique that takes labeled data samples and generates a model/classifier, that classifies new data samples into different predefined groups or classes. Over the years, many approaches have been developed [3] and they can be classified into different groups such as: (1) statistical approaches like linear discriminant analysis and multiple discriminant analysis, (2) decision tree approaches like C4.5 ,ID3, J48, and classification and regression tree (CART), (3) rule mining approaches like: Decision Table, JRip, oneR, zeroR, and PART and, (4) neural network approaches like multi-layer perceptron (MLP), learning vector quantization (LVQ), and radial basis function neural networks (RBFNs).

The interpretability or transparency is an important characteristic of a classifier. In this regard, rule-based classifiers became popular as the rules are easily understood by human being. Formally, classification rule mining problem can be defined as: assigning a class label C_j from a predefined set $C = \{C_1, C_2, \ldots, C_n\}$ to an object in a feature space S. The central idea is to find a function

$$F : S \rightarrow C, \tag{1}$$

which maps the unknown/unseen features/samples to a predefined class label. In case of numerical/continuous data, the crisp classification rules are not well suited; therefore, before classification it is recommended to partition and discretize these data. In practice, sharp boundary problem arises, by which some potential rules may be lost.

Fuzzy system has the powerful capability of handling uncertainty and vagueness, and hence, is suitable to solve this problem, and offers soft decision mechanism in classification system. As a result, a sample can belong to several classes with different degrees of membership value. Moreover, fuzzy rules are easy to understand, verify, and can be extended.

In fuzzy classification rule mining process, the identified tasks are: reduction in number of fuzzy classification rules, reduction in rule length, automatic generation of membership functions, automatic generation of fuzzy classification rules, and accuracy of the fuzzy classification rules. In lieu of these tasks, fuzzy classification rule mining problem is modeled as an optimization and search problem. The exponential increase in the number of rules and number of conditions in the antecedent part of the rules are major concern in handling high-dimensional problems with fuzzy rule-based system.

Evolutionary algorithms perform global search and cope better with attribute interaction [4], compared with greedy algorithms, hence used in the mining of high-level prediction rules and subsequently in fuzzy classification rule mining problems.

In broader sense, the major concerns of fuzzy classification rule mining problems are accuracy and interpretability. As these objectives are contradictory to each other, the desired solutions are treated as non-dominated, i.e., a Pareto front may be obtained for representing the solution space. So in place of single-objective genetic algorithms, multi-objective genetic algorithms are employed in finding solutions with a good trade-off between these objectives.

Evolutionary algorithms employed in real-world applications require complex models or computations in fitness calculation of an individual. In particular, in case of multiple objective optimization models, due to the curse of dimensionality problem, it requires more time in evaluating these individuals, which may lead to slower convergence of the algorithm. Fitness inheritance technique is used as a remedial measure, originally proposed by Smith et al. [5], for improving the performance of genetic algorithms. This technique allows reducing the number of real fitness function evaluations by substituting them with a surrogate, based on estimation of the finesses of parent individuals. In this work, we use a popular multi-objective genetic algorithm known as non-dominated sorting genetic algorithm-II (NSGA-II) for discovering the fuzzy rules with multiple criterions. To reduce the number of required fitness evaluations in NSGA-II, we use radial basis function neural networks (RBFNs) surrogate model to approximate the Pareto front of computationally expensive multi-objective fuzzy rule-based systems. The integration of RBFN and NSGA-II for discovering a small number of linguistically interpretable fuzzy rules is known as RBFN + NSGA-II in short.

Rest of the paper is organized as follows. Section 2 describes the basic materials. Section 3 describes multi-objective fuzzy genetic algorithms for classification rule mining. The proposed work is described in Sect. 4. Experimental work and conclusions are presented in Sects. 5 and 6, respectively.

2 Basic Materials

In data mining context, classification is performed through a two-step process. A classification model is built for describing a given set of data with known class labels in the first step [1], which is the learning step or training phase. In practice, the dataset/patterns are randomly divided into two mutually exclusive and exhaustive subsets, called training and test subsets. In the second step, the classification ability of the model is evaluated using the test subset. Classification accuracy is the basic criterion used in testing the

goodness of the model. With a satisfactory classification ability, the model is used for classifying future/unseen data. Beside this, various criteria like: speed, robustness, scalability, interpretability, support, confidence, coverage, undesirability, novelty, precision, recall, sensitivity, and specificity are used in measuring the goodness of classification rules.

2.1 Genetic Algorithms in Fuzzy Classification Rule Mining

In fuzzy classification rule discovery, genetic algorithms is applied extensively [6] as they search for solutions globally in state space, which increases the probability of convergence to the global optimal solution. Genetic algorithms carry out successfully the associated tasks involved in fuzzy classification rule mining such as: generation of fuzzy rule bases, optimization of fuzzy rule bases, generation of membership functions, and tuning of membership functions by modeling it as either search process or optimization process [7].

In an M-class problem, with n attributes and N labeled patterns, the fuzzy classification rule mining problem aims at generating a set of interpretable and accurate rules. The rules are either of the following two forms as in Eq. (2) or (3).

$$\text{Rule } R_q: \text{ IF } x_1 \text{ is } A_{q_1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{q_n} \text{ THEN Class } C_q, \quad q = 1, 2, \ldots, M \tag{2}$$

$$\text{Rule } R_q: \text{ IF } x_1 \text{ is } A_{q_1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{q_n} \text{ THEN Class } C_q, \text{ with } CF_q \quad q = 1, 2, \ldots, M \tag{3}$$

where R_q is the rule label of qth fuzzy if–then rule, $x = (x_1, x_2, \ldots, x_n)$ is an n-dimensional pattern vector, $A_{q_1}, A_{q_2}, \ldots, A_{q_n}$ are linguistic terms which represent the antecedent fuzzy sets, C_q is the class label, and CF_q is the rule weight, which is a value in [0, 1]. M is the total number of fuzzy rules in the rule base.

Generally, fuzzy classification rules are generated by a three step process. In the first step, each quantitative attribute is normalized into [0, 1]; in the second step, data are partitioned into fuzzy subspaces, by using k-membership functions [8] for k-linguistic values of each attribute; and in the third step, fuzzy rules are generated. Tree partition, grid partition, and hyper-box partition are popularly used partitioning techniques. In this paper, grid partitioning approach is followed.

Genetic algorithms [9] are population-based stochastic algorithms with highly intrinsic parallelism. Having the features like simplicity, versatility, robustness, efficacy, and efficiency, genetic algorithms are used to solve different optimization problems and in machine learning problems. Generally, genetic algorithms are suitable for the problems with a very big number of solutions, problems which need near-optimal solutions quickly, highly nonlinear systems and problem which need many near-global optimal solutions. Since fuzzy classification rule mining problem is having all these characteristics, this work realizes the importance of genetic algorithm.

A standard genetic algorithm is characterized by the following: a population of discrete structures/individuals representing candidate solutions for the problem being solved, a selection mechanism based on the aptitude/fitness of each individual, relative to the population and a set of idealized genetic operators that modify the individuals to create new genetic material.

The practices followed for genetic learning in rule mining process can be divided into three different categories based on their chromosome coding procedure [10] such as: Chromosome = Rule, Chromosome = Rule Set, and Chromosome = Decision Tree or Tree Rule. In "Chromosome = Rule" scheme, each rule is encoded in a single chromosome and the whole population evolves to the final rule set. This is again subdivided into three types: Michigan approach [11], Iterative Rule Learning (IRL) approach, and Genetic Cooperative Competitive Learning approach (GCCL). Michigan approach is also known as learning classifier system. Here the rule set is successively updated via one-by-one examination of training samples. It uses reinforcement learning and the GA, to learn rules which guide its performance in a given environment. It requires less memory and fewer evaluations. The IRL approach considers the best individuals as the solution, and the global solutions are figured by the best individuals obtained in multiple runs of the algorithm. In fact, the genetic algorithm/EA learns rule by rule iteratively and discards the samples from training data which is covered by each newly generated rule, following a divide-and-conquer approach. The GCCL approach codifies the rule base using the whole population or a part of it. A specified mechanism for maintaining diversity among the population is followed, which prevent the individuals for converging to same area of search space. The best fitted rules faced a competition to be in the final population. In "Chromosome = Rule set" scheme, each individual represents a whole rule set. This approach is known as Pittsburg approach [12]. This approach maintains a population of candidate rule sets whose quality is evaluated with a fitness function which consider different aspects of rule sets. In "Chromosome = Decision Tree or Tree Rule" scheme, decision trees are combine with genetic algorithms. The search potential of GA is used for finding highly accurate final tree. The tree is interpreted as a rule set, formed by the disjunction of the rules

resulted in each branch of the tree. Hence, the chromosome can encode the full tree or just a tree node.

2.2 Surrogate Models

Surrogate model/meta-model/approximation models [13] are mathematical or statistical models that imitate the behavior of a computationally expensive simulation code over the entire parameter space as accurately as possible, using the least number of data points. In particular, surrogate models are built to approximate computationally expensive simulation codes in less running time and to be used as an alias in exact analysis. If the true nature of a computer analysis code is represented as:

$$y = f(x), \tag{4}$$

then a surrogate model is an approximation of the form:

$$\hat{y} = \hat{f}(x) \quad \text{such} \quad \text{that} \quad y = \hat{y} + \epsilon, \tag{5}$$

where x is an input, y is an output variables, and ϵ is the error of approximation.

The motivation behind the use of meta-modeling techniques in evolutionary optimization algorithms is to substitute computationally expensive objective function evaluations with a computationally cheap, data-driven approximation of the original objective function [14]. In practice, the typical situations like: (1) fitness evaluation being highly costly, (2) absence of explicit model of fitness function, (3) requirement of supplementary fitness evaluations may be either in case of noisy fitness functions or in case of healthy solutions search, and (4) complication fitness function design, demand the use of an approximated fitness function .

An ideal comprehensive approximation of the original fitness function may not achieve easily due to shortage of data and high dimensionality of input space. In order to manage these situations, in one way: both original fitness function and the approximate model should be jointly used and in other way: for a given restricted number of data, the quality of the approximate model should be enhanced, where the quality of the model depends on model selection, online and off-line active data sampling and weighting, selection of training methods, and selection of error measures.

Approximate fitness models can be used in population initialization, reduction in randomness in crossover and mutation and fitness evaluations in evolutionary computation.

The popular meta-models used for the purpose are [15]: (1) polynomial models, (2) kriging models/design and analysis of computer experiment models/Gaussian regression models, (3) radial basis function neural network models, (4) multi-layer perceptron neural network models, and (5) support vector machine models.

The incorporation of approximate models constructed using historical data in evolutionary computation can be seen as one of the methods for incorporating knowledge into evolutionary systems [14]. Very interestingly approximate models have been embedded in almost every evolutionary algorithm, including migration initialization, recombination, mutation, fitness evaluations. Following three models are popularly used in the literature:

- *Use of approximate fitness evaluations through migration* The island model-based architecture has been proposed to incorporate information from approximate models to speed up the evolutionary algorithm [16]. For each level of approximation, one subpopulation has been introduced. Generally, each population evolves using its own level of approximation. The individuals in the subpopulation which uses higher accuracy approximation will be migrated into those using approximation of lower accuracy. Sefrioui and Periaux [17] proposed an extension to this model, where individuals can migrate from subpopulations using approximations of higher accuracy to subpopulations using approximate models of lower accuracy and vice versa.
- *Use of approximate fitness models for initializing the population and for guiding crossover and mutation* [18] Using approximation models in population initialization and guidance of genetic operators may reduce the risk of misleading the search direction instead of using it directly in fitness evaluations [19]. The logic behind this is initializations and genetic operations guided by an approximation model even with lower accuracy should usually be better than them randomly. But there is no significant reduction in fitness evaluations.
- *Use of approximate fitness models through fitness evaluations* In the recent trend, in order to reduce the fitness calculations, the approximate model has been used directly in fitness evaluation [20]. The approximation models like polynomials, kriging, and neural networks have been applied. A confidence interval for the fitness estimation is calculated as proposed in [21], to modify the model prediction so that the search in unexplored regions is encouraged. In case the original fitness function which is multimodal, the strategy leads to better performance.

3 Multi-objective Fuzzy Genetic Algorithms for Classification Rule Mining

The objectives considered in fuzzy classification rule mining are generally conflicting to each other, which necessitates the trade-off analysis on the solutions. Evolutionary multi-objective genetic algorithms specifically multi-objective genetic algorithms generate a large number of non-dominated solutions in a single run. Thus, user can perform trade-off analysis by observing all possible solutions.

The understanding of fuzzy rules become hard with more antecedent conditions [22]; hence, "don't care" is introduced as a fuzzy set in the antecedent part of the rule (Eq. 3). Fuzzy rules with many "don't care" conditions can be expressed in brief even for a high-dimensional pattern classification problem by omitting this "don't care" variable without any loss. For example, the fuzzy rule:

IF x_1 is small and x_2 is medium and x_3 is don't care THEN Class C_2 with CF_2 (6)

can be written as

IF x_1 is small and x_2 is medium THEN Class C_2 with CF_2 (7)

without loss of generality, where small and medium are fuzzy linguistic variables. In a fuzzy rule (Eq. 3), each antecedent is either any of k-linguistic variables/fuzzy sets (if k-variables are used) or don't care, total number of possible combinations of antecedent fuzzy sets is $(k+1)^n$. In low-dimensional problem, all these combinations can be inspected for rule generation and generated fuzzy rules corresponding to these are used as candidate rules, and hence, no pre-screening is required, but in high-dimensional problems some pre-screening methods are needed to apply in rule selection as the total number of possible combinations exponentially increases with the dimensionality of pattern space.

If S is the set of $(k+1)^n$ fuzzy rules for the n-dimensional classification problem, then the problem is to find a few simple fuzzy rules with high classification ability from the total rule set S. That is, a compact rule set RS with high classification accuracy is to be searched from the total rule set S.

The classification performance of the rule set RS ($RS \subset S$) is measured by the number of correctly classified training patterns. On the other hand the number of linguistic rules in RS and the total number of antecedent conditions in RS are used for measuring compactness. In the counting of antecedent conditions, "don't care" conditions are excluded. The total number of antecedent conditions in a rule is referred to as rule length.

Hence, the fuzzy rule mining problem can be formulated as:

$$\text{Maximize } f_1(RS), \text{minimize } f_2(RS), \text{ and minimize } f_3(RS), \tag{8}$$

where $f_1(RS)$ is the number of correctly classified training patterns by RS, $f_2(RS)$ is the number of rules in RS, and $f_3(RS)$ is the total rule length in RS.

The first objective $f_1(RS)$ is calculated by classifying all the m-training patterns by the rule set RS using the heuristic rule generation [7] and reasoning process described below. The consequent clause C_q of the fuzzy rule in (Eq. 3) is determined using a three step process.

Step 1 For each training pattern $x_q = (x_{q_1}, \ldots, x_{q_n})$, calculate the compatibility grade $\mu_{R_q}(x_q)$ with the rule R_q, using the formula:

$$\mu_{R_q}(x_q) = \mu_{q_1}(x_{q_1}) \times \cdots \times \mu_{q_n}(x_{q_n}), q = 1, 2, \ldots, m, \tag{9}$$

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

Step 2 For each class, calculate the total compatibility grade of the training patterns with the rule R_q. i.e.,

$$\alpha_{\text{class h}}(R_q) = \sum_{x_q \in \text{class h}} \mu_{R_q}(x_q), h = 1, 2, \ldots, M. \tag{10}$$

Step 3 The consequent class C_q is the Class $h, h = 1, 2, \ldots M$ which has the maximum value of $\alpha_{\text{class h}}(R_q)$., i.e.,

$$\alpha_{\text{class C}_q}(R_q) = \max\{\alpha_{\text{class 1}}(R_q), \alpha_{\text{class 2}}(R_q), \ldots, \alpha_{\text{class M}}(R_q)\} \tag{11}$$

Step 4 The grade of certainty CF_q of the fuzzy rule R_q is 0 (zero) if consequent class C_q is ϕ. Otherwise it is calculated using the formula:

$$CF_q = \frac{\left(\alpha_{\text{Class h}}(R_q) - \bar{\beta}\right)}{\sum_{h=1}^{M} \alpha_{\text{Class h}}(R_q)}, \tag{12}$$

where $\bar{\beta} = \sum_{h=1, h \neq h_q}^{M} \frac{\alpha_{\text{Class h}}(R_q)}{(M-1)}$.

If the consequent clause cannot be determined uniquely, the rule R_q is discarded. In particular if $\alpha_{\text{class h}}(R_q) = 0$, then rule R_q is not generated. After the generation of the rule set RS using the procedure defined above, an unknown pattern $x_q = (x_{q_1}, \ldots, x_{q_n})$ is classified by the rule set RS using the reasoning method based on a single winner rule [23]. The single winner rule $R_v \in RS$ is determined for a new pattern $x_p = (x_{p_1}, \ldots, x_{p_n})$ as follows:

- For all rule R_q in RS, calculate compatibility grade $\mu_{A_q}(x_p) \times CF_q$, where A_q is the antecedent of rule R_q.
- Find the maximum of $\mu_{A_q}(x_p) \times CF_q$. The antecedent A_q for which $\mu_{A_q}(x_p)$ is maximum is the antecedent of winner rule. i.e., A_v.
- The new pattern x_p is classified as Class C_v, which is the consequent clause of winner rule R_v, with antecedent A_v. If multiple fuzzy rules have equal $\mu_{A_q}(x_s) \times CF_q$ value but different consequent clasue, then x_p is left as unclassified. If no fuzzy rules is compatible with x_p, then also classification is rejected.

In practice, there is no optimal rule set which satisfy all the three objectives. Hence, the task is to find multiple rule sets that are non-dominated. A rule set RS_2 is said to dominate another rule set RS_1, i.e., RS_2 is better than RS_1 if all of the following inequalities hold

$$f_1(RS_1) \leq f_1(RS_2), f_2(RS_1) \geq f_2(RS_2), f_3(RS_1) \geq f_3(RS_2) \tag{13}$$

and at least one of the following inequalities holds

$$f_1(RS_1) < f_1(RS_2), f_2(RS_1) > f_2(RS_2), f_3(RS_1) > f_3(RS_2) \tag{14}$$

The first condition infers that no objective of RS_2 is worse than RS_1. The second condition means that at least one objective of RS_2 is strictly better than RS_1. The rule set RS is said to be a Pareto-optimal solution if it is non-dominated. Multi-objective genetic algorithms are used to find these non-dominated rule sets or Pareto-optimal solutions. Generally, following procedure is adopted for the purpose.

An arbitrary subset RS of the n candidate rules is represented as a binary string of length n as

$$RS = rs_1 rs_2 \ldots rs_n, \tag{15}$$

where $rs_q = 1$ implies the inclusion of qth candidate rule R_q and $rs_q = 0$ implies the non-inclusion of qth candidate rule R_q in RS. Randomly an initial population of pre-specified number is generated having binary strings of length n.

Classifying all the given training patterns by RS, the first objective $f_1(RS)$ of each string RS is calculated. The classification is performed by finding a single winner rule for each training pattern, as single winner rule is adhered. Hence, some rules may not win and hence can be removed from the rule set RS, without degrading the classification accuracy. In parallel, the second and third objectives are also improved by removing the weaker rules. In the process, all the fuzzy rules that are non-winners are removed from the rule set RS. Changing the 1 bit to 0's, in each string of the current population, removal of unnecessary rules is done. Any rule from all the strings in the current population cannot be removed without examining each string, as some rules without any contribution in one rule set may have great contribution in another rule set, due to the combinatorial nature of this problem. After the removal of unnecessary rules, the second and third objectives are computed.

After the calculation of the three objectives of each string in the current population, the secondary population is updated, so that it includes all the non-dominated strings among the examined strings. That is, each string of the current population is examined, whether it is dominated by other strings of the current population and secondary population. If it is not dominated by any other string, a copy of it is added to secondary population. Then, all strings dominated by the newly added string are removed from secondary population. In every generation, secondary population is updated in this process. The fitness value of each string or rule set RS in the current population is defined by the three objectives as:

$$\text{fitness}(RS) = w_1 \times f_1(RS) - w_2 \times f_2(RS) - w_3 \times f_3(RS), \tag{16}$$

where w_1, w_2, and w_3 are nonnegative weights satisfying $w_1 + w_2 + w_3 = 1$. These weights are selected randomly.

The newly generated strings by selection, crossover, and mutation constitute the next population. Some non-dominated strings may be selected from the secondary population as elite solution, and their copies added to the new population. Then, again genetic operators are applied to generate new population and this process is continued until the satisfaction of the termination condition is satisfied. The non-dominated strings (i.e., rule sets) are obtained from the secondary population as our required solution.

Classically, weighted sum approach is used to optimize these objectives, by assigning nonnegative weight to each objective [e.g., Eq. (16)]. Ishibuchi et al. [24] used weighted sum approach using two objectives, minimizing the number of correctly classified training patterns (as accuracy measure) and minimizing the number of fuzzy rules (as complexity measure) in solving fuzzy classification rule mining problems. Difficulty with this method is the specification of appropriate weight to the objectives which are problem dependent and the final solution depends on the weight specification.

In order to resolve this difficulty, multi-objective approaches like NSGA-II [25], improved strength Pareto evolutionary algorithm (SEPA2), [26], neighborhood cultivation genetic algorithm (NCGA) [27], and non-dominated rank-based sorting genetic algorithm (NRSGA) [28] are used. In this work, NSGA-II is employed in fuzzy classification rule mining problems to search for a number of non-dominated solutions with different trade-offs

between accuracy and complexity. Multi-obejective genetic algorithms able to generate trade-off between interpretability and accuracy [29, 30]. Some of the advancement in multi-objective fuzzy classification rule mining method (FCRM) can be obtained in [31–46].

4 Proposed Work

The proposed approach of solving multi-objective fuzzy classification problem is based on surrogate-assisted models. We used the surrogate RBF networks synergistically combined with NSGA-II. In Sects. 4.1 and 4.2, we describe the integration of RBFNs + NSGA-II and its applications in fuzzy classification, respectively.

4.1 RBFNs + NSGA-II

The RBFNs model adopted here to build the surrogate model is shown in Fig. 1. Recall that the role of the surrogate model is to reconstruct the real function that maps the design variables to the space of objectives. Local reconstruction of maps, even though it has to be repeated for each new member of the population, seems to provide better approximations than trying to build a global metamodel. In addition, RBFNs incorporate ingredients from multivariate interpolation, regularization theory and enjoy many desirable properties, the most important being that of universal approximation. Radial basis functions (RBFs) comprises of two terms RBFs. A radial function is a function of the type:

$$f : \mathbb{R}^d \rightarrow \mathbb{R} : (x_1, \ldots, x_d) \mapsto \phi(||x_1, \ldots, x_d||_2), \tag{17}$$

for some function $\phi : \mathbb{R} \rightarrow \mathbb{R}$.

This implies that value of the function f at any point $\vec{x} = (x_1, \ldots, x_d)$ only depends on the Euclidean norm of $\vec{x}$, which is calculated using the following formula:

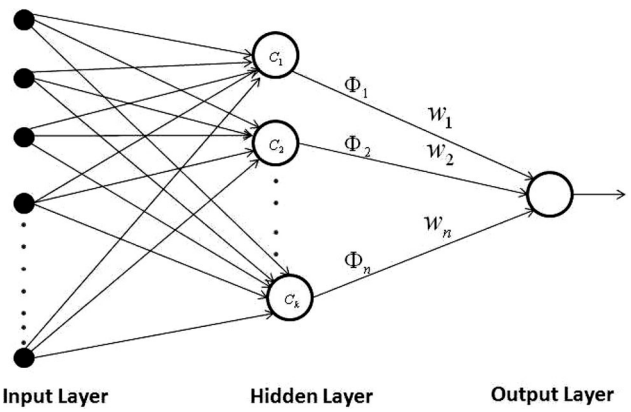


Fig. 1 Radial basis function neural networks

$$||\vec{x}||_2 = \sqrt{\sum_{j=0}^{d} x_j^2} \tag{18}$$

This explains the term radial. Let us suppose we have some points $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n \in \mathbb{R}^d$ called centers. The linear combination of the function f centered at the points $\vec{x}$ is given by:

$$g : \mathbb{R}^d \mapsto \mathbb{R} : \mapsto \vec{x} \mapsto \sum_{j=1}^{n} \lambda_i f(\vec{x} - \vec{x}_i) = \sum_{j=1}^{n} \lambda_j \phi(||\vec{x} - \vec{x}_j||), \tag{19}$$

where $||\vec{x} - \vec{x}_j||$ is the Euclidean distance between the points $\vec{x}$ and $\vec{x}_j$. So g becomes a function which is in the finite dimensional space spanned by the basis functions:

$$f_j : \vec{x} \mapsto g(||\vec{x} - \vec{x}||) \tag{20}$$

Let us have the values of a certain function $H : \mathbb{R}^d \mapsto \mathbb{R}$ at a set of fixed locations $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$. These values are $g_i = H(\vec{x}_i)$, so $\vec{x}_i$ are used as centers in Eq. (19). If we want to force the function g to take values g_i at the different point $\vec{x}_i$, then we have to put some condition on λ_j. This implies $\forall i \in 1, \ldots, n, g_i = g(\vec{x}_i) = \sum_{j=1}^{n} (\lambda_j . \phi(||\vec{x}_i - \vec{x}_j||)$.

Linear spline, cubic spline, multi-quadrics spline, thin-plate spline, and Gaussian function are the commonly used basis functions as shown in Table 1. A Gaussian kernel is a good choice.

An input layer having equal number of inputs with the problem dimension, a single hidden layer with k nonlinear processing units, and an output layer with linear weights w_j comprise a RBF network as described in Fig. 1. In case of small sample, the size of the hidden layer can be equal to the sample size; however, in case of big sample, the size of hidden layer should be smaller than the sample size, in order to reduce the number of calculations. The output $p(x)$ of the RBF network is a linear combination of RBF $\phi(.)$, defined as:

$$p(x) = w_0 + \sum_{j=1}^{m} w_j \phi_j(||x - c_j||), \tag{21}$$

Table 1 Commonly used radial basis function

Name	Definition
Linear spline	$\|r\|$
Cubic spline	$\|r\|^{2m+1} \ln \|r\|$
Multi-quadric	$\|r\|^3$
Thin-plate spline	$\sqrt{1 + (\epsilon r)^2}$
Gaussian function	$e^{-(\epsilon r)^2}$

where w_0 and w_j are the unknown coefficients to be calculated. ||.|| is the Euclidean norm, $\phi_j(||x - c_j||)$ represents the jth RBF, also called kernel, which evaluates the distance between the input x and the center c_j. The center c_j is an unknown and hence learned by methods such as k-means clustering method.

In multi-objective optimization, the points in the design space that constitutes the Pareto front which are spread over an extended region. This makes inadequate use of an RBFN that only interpolates the training patterns. Such a network "learns" perfectly the details of the input–output mapping in the close vicinity of the training patterns, but usually fails to generalize, i.e., it generally estimates poorly the objective functions for a new individual that is located in the intermediate space between the training patterns. To overcome this difficulty, the centers of the RBFN must be less than the training patterns, such that the later be approximated instead of interpolated.

As can be seen in NSGA-II, which is used to optimize the approximated model generated by the RBFNs. Our approach keep two populations; the main population which is used to select the parents and a secondary population that retains dominated solutions found during the evolutionary process.

First, we generate P individuals using Latin hypercubes which is a method that guarantees a good distribution of the initial population in a multidimensional space. If we do a simple random sampling of the initial points, where the new sample points are generated without taking into account the previously generated sample points, we may not be able to obtain points in the same critical areas from the search space. Our approximation model requires a good distribution of the sample points provided, in order to build a good approximation of the discriminating functions and therefore the importance of adopting this approach has been realized. A Latin cube is a selection of one point from each row and column of a square matrix. In M dimensions, the corresponding item is a set of P points, where in each dimension there is exactly one point per column or range of values in M dimensions, these objects are called Latin hypercubes. Once a Latin-Hypercube is created, we choose the center of each hypercube as the place where the initial P individuals are chosen. Then, we evaluate P individuals with the real objective functions and train the meta-model using the RBFNs. As we are dealing with multiple objective problems, we decided to train the multiple objectives separately. Consequently, we obtain a different RBFN per each objective. So these objectives are still in conflict with each other as they are an approximate model of the real objectives. Thus, now we have to solve a different multi-objective problem based on the different RBFN obtained during the training process.

We have used NSGA-II, which adopts a fast non-dominated sorting approach to classify solutions according to the level of dominance and a crowding distance operator, which is responsible for preserving diversity. The NSGA-II is adopted to optimize the meta-model obtained by the RBFNs. From all the non-dominated solutions found by the NSGA-II, we decided to retain only 20 individuals. These new individuals obtained by the NSGA-II are compared with respect to all the points in the main population and chosen that are different with respect to all the points contained in the main population that are accepted and evaluated using the real objective function values. All the solutions contained in the main population are used to retrain the meta-model (using RBFNs) and got another approximation of the real objectives. This procedure is repeated until the number of maximum evaluations is performed. With this procedure, the size of the main population is increased as the real objective function evaluation is performed. As the main population become larger, the training process takes more computational time to do the approximation because the ϕ matrix is larger and the matrix inversion process takes more and more time. So we decided to accept a maximum of 500 solutions in the main population. When this number is reached, we choose only the 300 best solutions (based on rank and crowding distance sorting) to continue with the training process. At the end of the procedure, we select 52 non-dominated solutions from the main population, and we store 100 dominated solutions in the secondary population. Every removed point from the main population is included in the secondary population. If this secondary population reaches a size of 100 points, a rank and crowding distance sorting is used to keep only 100 points. (The remaining points are eliminated.)

4.2 RBFNs + NSGA-II in Fuzzy Classifiers

We use RBFNs-assisted NSGA-II for finding non-dominated rule sets. We formulate our task of designing comprehensible fuzzy rule-based classification systems for the following three-objective optimization problem.

$$\text{Maximize } f_1(RS), \tag{22}$$

$$\text{Minimize } f_2(RS), \tag{23}$$

$$\text{Minimize } f_3(RS), \tag{24}$$

where $f_1(RS)$ is the number of correctly classified training patterns by a rule set in RS, $f_2(RS)$ is the number of fuzzy rules in RS, and $f_3(RS)$ is the total number of antecedent conditions in RS.

Now our task is to find non-dominated rule sets with respect to the three objectives. Let us briefly explain the

concept of non-dominated rule sets for our three-objective optimization problem.

A rule set RS is said to be dominated by another rule set RS^* if all the following inequalities hold:

$$f_1(RS) \leq f_1(RS^*), f_2(RS) \geq f_2(RS^*), f_3(RS) \leq f_3(RS^*), \tag{25}$$

and at least one of the following inequalities holds:

$$f_1(RS) < f_1(RS^*), f_2(RS) > f_2(RS^*), and\ f_3(RS) > f_3(RS^*). \tag{26}$$

The first condition (i.e., all three inequalities) means that no objective of RS^* is worse than RS. The second condition (i.e., one of the three inequalities) means that at least one objective of RS^* is better than RS. If there exists no RS^* that satisfies the above two conditions, the rule set S is said to be a non-dominated rule set.

Let us discuss how our method creates non-dominated rule sets from all combinations of antecedent linguistic values. Our algorithm is a combination of two approaches, Michigan approach, and Pittsburgh approach.

Encoding/Individual Representation Since the consequent class and the certainty grade can be easily determined by the procedure described in Sect. 3 from the given training patterns, our algorithm is used for determining the number of fuzzy rules and the antecedent part of each rule. Thus, only the antecedent part is coded as a string. A rule set RS is denoted by a concatenated string.

5 Experimental Study

This section is divided into three Sects. 5.1, 5.2, and 5.3. In Sect. 5.1, the description of the datasets is presented. The setting of parameters, environment, and evaluation matrices is presented in Sect. 5.2 . Simulation results and analysis are reported in Sect. 5.3.

Table 2 Description of Iris dataset

Attribute	Type	Domain	Description
1	Real	[4.3, 7.9]	Sepal length in cm
2	Real	[2.0, 4.4]	Sepal width in cm
3	Real	[1.0, 6.9]	Petal length in cm
4	Real	[0.1, 2.5]	Petal width in cm

5.1 Dataset Description

To validate the methods, we have taken three datasets IRIS, WINE, and GLASS datasets and taken from KEEL data repository.

Iris Dataset Iris dataset is a three-class problem, consisting of 150 features. It consists of 50 from each Class 1 (*Iris setosa*), Class 2 (*Iris versicolor*) and Class 3 (*Iris virginica*). Table 2 presents the detailed description of the attributes of the dataset.

Wine Dataset The chemical analysis of three types of wine resulted this dataset. It contains the magnitude of 13 types of ingredients present in all three types of wine. There are 59 instances from Class 1, 71 instances from Class 2, and 48 instances from Class 3. Table 3 presents the detailed description of the attributes of the dataset.

Glass Dataset Glass Identification Database classification was motivated by criminological investigation. The glass left at the sight of the crime is used as evidence (if it is correctly identified) by the chemical analysis of 7 different types of glass. There are 10 attributes including one identification number (Id number). In 214 number of instances, 70 are float-processed building-window glass (Class 1), 17 are float-processed building-window glass (Class 2), 76 are non-float-processed building-window glass (Class 3), none from non-float-processed vehicle-window glass (Class 4), 13 container glass (Class 5), 9 tableware glass (Class 6), and 29 headlamps glass (Class 7). Table 4 presents the detail description of the attributes of the dataset.

5.2 Environment, Parameter Setting, and Evaluation Matrices

All the experiments have been performed by using the following hardware and software configurations. An Intel core(TM)i5-4440s with CPU @2.80 GHz and 4 GB RAM along with windows 10, 64 bit operating system has been used as the basic hardware and software configurations. MATLAB R2017a has been used for validating the proposed surrogate-assisted model.

Our proposed algorithm is executed with the parameters setting like: (1) population size = 50, (2) crossover probability = 0.9, (3) mutation probability = 0.6.

The method was applied to each dataset 30 times.

Since we had no domain knowledge on each dataset, we use the five linguistic values with the triangular membership function (shown in Fig. 2) for every attribute of each set.

The classification performance of all the models is based on the evaluation matrices derived from a confusion matrix or contingency table. It is a two-by-two matrix that holds the possible outcomes for a given classifier and a set of

Table 3 Attribute description of Wine dataset

Attribute name	Type	Domain	Description
Alcohol	Real	[11.0, 14.9]	Alcohol
Malic-acid	Real	[0.7, 5.8]	Malic acid
Ash	Real	[1.3, 3.3]	Ash
Alkalinity-of-ash	Real	[10.6, 30.0]	Alkalinity of ash
Magnesium	Real	[70.0, 162.0]	Magnesium
Total-phenols	Real	[0.9, 3.9]	Total phenols
Flavonoids	Real	[0.3, 5.1]	Flavonoids
Nonflavonoid-phenols	Real	[0.1, 0.7]	Nonflavonoid phenols
Proanthocyanidins	Real	[0.4, 3.6]	Proanthocyanidins
Color intensity	Real	[1.2, 13.0]	Color intensity
Hue	Real	[0.4, 1.8]	Hue
OD280/OD315	Real	[1.2, 4.0]	OD280/OD315 of diluted wines
Proline	Real	[278.0, 1680]	Proline

Table 4 Attribute description of Glass dataset

Attribute name	Type	Domain	Description
ID number	Integer	[1, 214]	Identification number
RI	Real	[1.5112, 1.53393]	Refractive index
Na	Real	[10.73, 17.38]	Sodium
Mg	Real	[0.0, 4.49]	Magnesium
Al	Real	[0.29, 3.5]	Aluminum
Si	Real	[69.81, 75.41]	Silicon
K	Real	[0.0, 6.21]	Potassium
Ca	Real	[5.43, 16.19]	Calcium
Ba	Real	[0.0, 3.15]	Barium
Fe	Real	[0.0, 0.51]	Iron

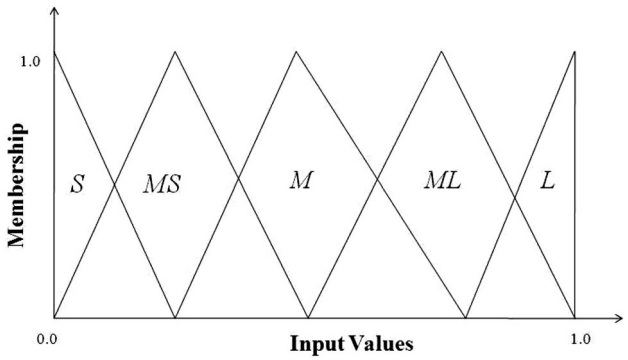


Fig. 2 Five linguistic values with triangular membership functions

instances. The four possible outcomes are true positive, true negative, false positive, and false negative. If the instance is positive and it is classified as positive, it is counted as true positive (TP). If the instance is positive and it is classified as negative, it is counted as false negative (FN). If the instance is negative and it is classified as negative, it is counted as true negative (TN). If the instance is negative and it is classified as false positive (FP). Table 5 shows the confusion matrix.

Table 5 Confusion matrix

Actual class	Predicted class	
	Positive	Negative
Positive	True positive	False negative
Negative	False positive	True negative

Following are some common matrices derived using confusion matrix:

$$\text{True Positive Rate (TP Rate)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{27}$$

$$\text{False Positive Rate (FP Rate)} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{28}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{TP} + \text{FP}} \tag{29}$$

$$F\text{-Measure} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{30}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{31}$$

5.3 Simulation Results

The simulation results obtained from our intensive experimental study have been presented in two subsections. Section 5.3.1 presents the non-dominated solutions obtained through RBF network surrogate and non-surrogate-assisted multi-objective genetic algorithms for fuzzy rule-based system. Section 5.3.2 presents a comparative

Table 6 Non-dominated solutions obtained from Iris data

Non-dominated set	RBFNs + NSGA-II		NSGA-II	
	Total length	Average length	Total length	Average length
1	1	1.0	1	1.0
2	2	1.0	2	1.0
3	3	1.0	3	1.0
4	4	1.0	4	1.0
5	5	1.4	5	1.4

Table 7 Non-dominated solutions obtained from Wine data

Non-dominated set	RBFNs + NSGA-II		NSGA-II	
	Total length	Average length	Total length	Average length
1	5	1.67	5	1.00
2	6	1.70	7	1.40
3	10	1.75	9	1.50
4	8	1.68	11	1.57
5	15	1.78	17	1.89

Table 8 Non-dominated solutions obtained from Glass data

Non-dominated set	RBFNs + NSGA-II		NSGA-II	
	Total length	Average length	Total length	Average length
1	3	1.00	3	1.00
2	5	1.25	5	1.50
3	9	1.80	6	1.80
4	14	1.75	14	1.70
5	20	2.00	20	2.00

analysis of only classification accuracy obtained through our models with five non-surrogate, non-fuzzy, and non-multi-objective rule mining approaches.

5.3.1 *Non-dominated Solutions from RBFN + NSGA-II and NSGA-II*

A set of non-dominated solutions is finally obtained after 1000th generations. From 30 trials, we obtained 30 sets of non-dominated solutions. For summarizing simulation results concisely, we merged them into a single solution set and compared solutions with each other. In such a comparison, some solutions were dominated by other solutions obtained from different trials. All solutions that were dominated by other solutions from different trials were removed from the enlarged solution set. The refined solution set is reported as simulation results of our algorithm for each dataset. Since the classification performance is measured by the number of correctly classified training patterns in our problem, we report the value of this objective of each non-dominated solutions together with the other two objective values.

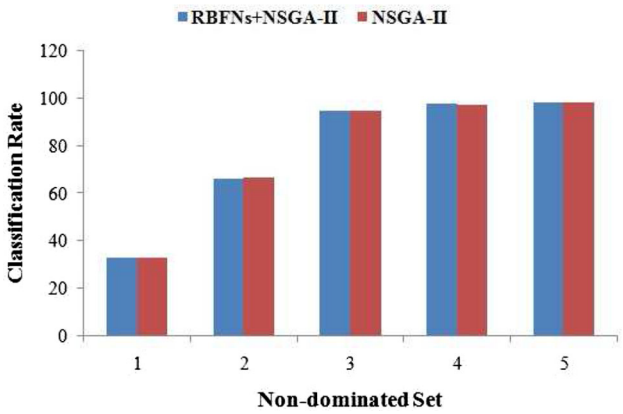


Fig. 3 Non-dominated sets versus classification rate of IRIS data

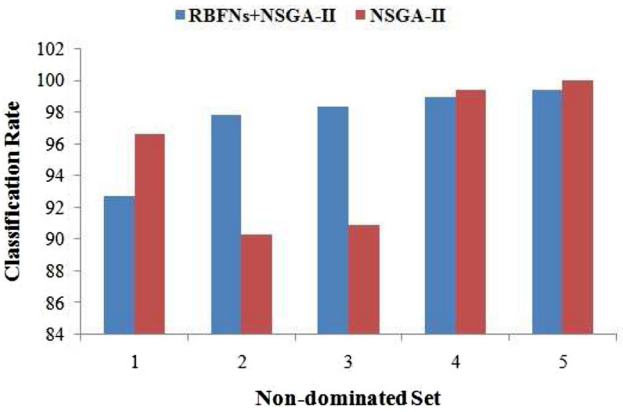


Fig. 4 Non-dominated sets versus classification rate of WINE data

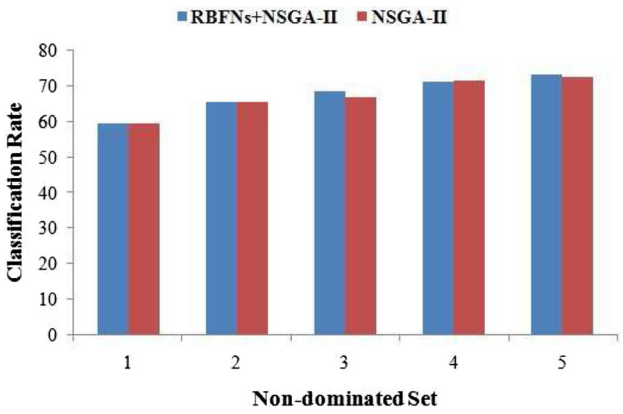


Fig. 5 Non-dominated sets versus classification rate of GLASS data

Table 9 Confusion matrix for Iris data using decision table

Iris setosa	Iris versicolor	Iris virginica
50	0	0
0	46	4
0	8	42

Table 10 Confusion matrix for Iris data using JRIP rules

Iris setosa	Iris versicolor	Iris virginica
48	2	0
0	46	4
0	4	46

Table 11 Confusion matrix for Iris data using OneR

Iris setosa	Iris versicolor	Iris virginica
50	0	0
0	46	4
0	8	42

Table 12 Confusion matrix for Iris data using PART

Iris setosa	Iris versicolor	Iris virginica
49	1	0
0	47	3
0	4	46

Table 13 Confusion matrix for Iris data using ZeroR

Iris setosa	Iris versicolor	Iris virginica
50	0	0
50	0	0
50	0	0

Table 14 Confusion matrix of Iris data using NSGA-II

Iris setosa	Iris versicolor	Iris virginica
50	0	0
0	47	3
0	2	48

Iris Dataset We use this dataset to illustrate the performance of our algorithm, while it is not actually a high-dimensional as well as complex classification problem. Fuzzy rules of the following type may be generated for IRIS dataset with 4 attributes.

Table 15 Confusion matrix of Iris data using RBFN + NSGA-II

Iris setosa	Iris versicolor	Iris virginica
50	0	0
0	47	3
0	1	49

Table 16 Classification performance w.r.t different evaluation measures for Iris data using decision table

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Iris setosa	1.000	0.000	1.000	1.000	1.000
Iris versicolor	0.920	0.080	0.852	0.885	0.825
Iris virginica	0.840	0.040	0.913	0.875	0.818
Weighted-average	0.920	0.040	0.922	0.920	0.881

Table 17 Classification performance w.r.t different evaluation measures for Iris data using JRIP rules

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Iris setosa	0.960	0.000	1.000	0.980	0.970
Iris versicolor	0.920	0.060	0.885	0.902	0.852
Iris virginica	0.920	0.040	0.920	0.920	0.880
Weighted-average	0.933	0.033	0.935	0.934	0.901

Table 18 Classification performance w.r.t different evaluation measures for Iris data using OneR

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Iris setosa	1.000	0.000	1.000	1.000	1.000
Iris versicolor	0.920	0.080	0.852	0.885	0.825
Iris virginica	0.840	0.040	0.913	0.875	0.818
Weighted-average	0.920	0.040	0.922	0.920	0.881

Table 19 Classification performance w.r.t different evaluation measures for Iris data using PART

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Iris setosa	0.980	0.000	1.000	0.990	0.985
Iris versicolor	0.940	0.050	0.904	0.922	0.882
Iris virginica	0.920	0.030	0.939	0.929	0.895
Weighted-average	0.947	0.027	0.948	0.947	0.920

Table 20 Classification performance w.r.t different evaluation measures for Iris data using ZeroR

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Iris setosa	1.000	1.000	0.333	0.500	0.000
Iris versicolor	0.000	0.000	0.000	0.000	0.000
Iris virginica	0.000	0.000	0.000	0.000	0.000
Weighted-average	0.333	0.333	0.111	0.167	0.000

Table 21 Classification performance w.r.t different evaluation measures for Iris data using NSGA-II

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Iris setosa	1.000	0.000	1.000	1.000	1.000
Iris versicolor	0.940	0.020	0.959	0.949	0.925
Iris virginica	0.960	0.030	0.941	0.950	0.925
Weighted-average	0.967	0.017	0.967	0.967	0.950

Table 22 Classification performance w.r.t different evaluation measures for Iris data using RBFN + NSGA-II

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Iris setosa	1.000	0.000	1.000	1.000	1.000
Iris versicolor	0.940	0.010	0.979	0.959	0.940
Iris virginica	0.980	0.030	0.942	0.961	0.941
Weighted-average	0.973	0.013	0.974	0.973	0.960

Table 23 Confusion matrix for Wine data using decision table

Class 1	Class 2	Class 3
56	3	0
13	58	0
0	7	41

Table 24 Confusion matrix for Wine data using JRIP rules

Class 1	Class 2	Class 3
55	4	0
4	65	2
0	2	46

Table 25 Confusion matrix for Wine data using OneR

Class 1	Class 2	Class 3
55	4	0
19	46	6
0	9	39

Table 26 Confusion matrix for Wine data using PART

Class 1	Class 2	Class 3
56	3	0
4	64	3
1	4	43

Table 27 Confusion matrix for Wine data using ZeroR

Class 1	Class 2	Class 3
0	59	0
0	71	0
0	48	0

Table 28 Confusion matrix of Wine data using NSGA-II

Class 1	Class 2	Class 3
59	0	0
1	69	1
0	0	48

Table 29 Confusion matrix of Wine data using RBFN + NSGA-II

Class 1	Class 2	Class 3
59	0	0
1	68	2
0	0	48

Table 30 Classification performance w.r.t different evaluation measures for Wine data using decision table

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	0.949	0.109	0.812	0.875	0.812
Class 2	0.817	0.093	0.853	0.853	0.729
Class 3	0.854	0.000	1.000	0.921	0.900
Weighted-average	0.871	0.073	0.879	0.871	0.803

Table 31 Classification performance w.r.t different evaluation measures for Wine data using JRIP rules

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	0.932	0.034	0.932	0.932	0.899
Class 2	0.915	0.056	0.915	0.915	0.859
Class 3	0.958	0.015	0.958	0.958	0.943
Weighted-average	0.933	0.038	0.933	0.933	0.895

Table 32 Classification performance w.r.t different evaluation measures for Wine data using OneR

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	0.932	0.160	0.743	0.827	0.738
Class 2	0.648	0.121	0.780	0.708	0.548
Class 3	0.813	0.046	0.867	0.839	0.783
Weighted-average	0.787	0.114	0.791	0.783	0.674

Table 33 Classification performance w.r.t different evaluation measures for Wine data using PART

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	0.949	0.042	0.918	0.933	0.900
Class 2	0.901	0.065	0.901	0.901	0.836
Class 3	0.896	0.023	0.953	0.915	0.870
Weighted-average	0.916	0.046	0.916	0.916	0.870

Table 34 Classification performance w.r.t different evaluation measures for Wine data using ZeroR

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	0.000	0.000	0.000	0.000	0.000
Class 2	1.000	1.000	0.399	1.000	0.570
Class 3	0.000	0.000	0.000	0.000	0.000
Weighted-average	0.399	0.399	0.159	0.227	0.000

Table 35 Classification performance w.r.t different evaluation measures for Wine data using NSGA-II

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	1.000	0.008	0.983	0.992	0.987
Class 2	0.972	0.000	1.000	0.986	0.977
Class 3	1.000	0.008	0.980	0.990	0.986
Weighted-average	0.989	0.005	0.989	0.989	0.983

Table 36 Classification performance w.r.t different evaluation measures for Wine data using RBFN + NSGA-II

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	1.000	0.008	0.983	0.992	0.987
Class 2	0.958	0.000	1.000	0.978	0.965
Class 3	1.000	0.015	0.960	0.980	0.972
Weighted-average	0.983	0.007	0.984	0.983	0.974

Table 37 Confusion matrix for Glass data using decision table

Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
58	8	4	0	0	0	0
19	53	3	0	1	0	0
11	1	4	0	0	0	1
0	0	0	0	0	0	0
3	3	0	0	6	0	1
1	5	0	0	1	2	0
3	5	0	0	1	0	20

Table 38 Confusion matrix for Glass data using JRIP rules

Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
48	19	1	0	0	1	1
11	56	3	0	2	1	3
8	8	0	0	0	1	0
0	0	0	0	0	0	0
0	3	0	0	8	0	2
0	2	0	0	1	6	0
0	6	0	0	1	0	22

Table 39 Confusion matrix for Glass data using OneR

Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
53	17	0	0	0	0	0
24	47	0	0	0	0	5
11	6	0	0	0	0	0
0	0	0	0	0	0	0
1	7	0	0	3	0	2
3	5	0	0	0	0	1
2	5	0	0	0	0	22

$$R_i = \text{IF } x_1 \text{ is } A_{i_1} \text{ and } \ldots \text{ and } x_4 \text{ is } A_{i_4} \quad \text{THEN Class } C_i \text{ with } CF_i. \tag{32}$$

Since there are only 4 attributes in this dataset, we can examine all the $(5+1)^4 = 1296$ combinations of antecedent linguistic values for generating candidate fuzzy

Table 40 Confusion matrix for Glass data using PART

Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
53	11	5	0	0	0	1
13	50	9	0	1	1	2
11	3	3	0	0	0	0
0	0	0	0	0	0	0
0	4	0	0	8	0	1
0	1	0	0	0	8	0
1	2	0	0	2	0	24

Table 41 Confusion matrix for Glass data using ZeroR

Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
0	70	0	0	0	0	0
0	76	0	0	0	0	0
0	17	0	0	0	0	0
0	0	0	0	0	0	0
0	13	0	0	8	0	0
0	9	0	0	0	8	0
0	29	0	0	0	0	0

Table 42 Confusion matrix for Glass data using NSGA-II

Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
57	12	1	0	0	0	0
18	50	1	0	2	4	1
10	7	0	0	0	0	0
0	0	0	0	0	0	0
0	1	0	0	11	0	1
1	1	0	0	1	5	1
1	3	0	0	1	1	23

Table 43 Confusion matrix for Glass data using RBFN + NSGA-II

Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
56	12	2	0	0	0	0
18	49	3	0	1	1	4
12	3	2	0	0	0	0
0	0	0	0	0	0	0
0	3	0	0	9	0	1
0	1	0	0	0	7	1
1	0	0	0	2	0	26

rules. By examining those combinations, we generate 507 fuzzy rules from the given 150 patterns. Some fuzzy rules could not be generated, because no training patterns were compatible with those rules. For obtaining non-dominated rule sets, we applied our algorithm 30 times using different initial populations. From these 30 times, we found 5 non-dominated sets with respective objective values as shown in Table 6. In this table, the column "average length" shows the average number of antecedent conditions of each fuzzy rule.

Table 44 Classification performance w.r.t different evaluation measures for Glass data using decision table

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	0.829	0.257	0.611	0.703	0.540
Class 2	0.697	0.159	0.707	0.702	0.540
Class 3	0.235	0.036	0.364	0.286	0.245
Class 4	0.000	0.000	0.000	0.000	0.000
Class 5	0.462	0.015	0.667	0.545	0.531
Class 6	0.222	0.000	1.000	0.364	0.464
Class 7	0.690	0.011	0.909	0.784	0.765
Weighted-average	0.668	0.146	0.685	0.657	0.543

Table 45 Classification performance w.r.t different evaluation measures for Glass data using JRIP rules

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	0.686	0.132	0.716	0.701	0.560
Class 2	0.737	0.275	0.596	0.659	0.445
Class 3	0.000	0.020	0.000	0.000	0.041
Class 4	0.000	0.000	0.000	0.000	0.000
Class 5	0.615	0.020	0.667	0.640	0.618
Class 6	0.667	0.015	0.667	0.667	0.652
Class 7	0.759	0.032	0.786	0.772	0.732
Weighted-average	0.654	0.149	0.621	0.635	0.503

Table 46 Classification performance w.r.t different evaluation measures for Glass data using OneR

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	0.757	0.285	0.564	0.646	0.447
Class 2	0.618	0.290	0.540	0.557	0.320
Class 3	0.000	0.000	0.000	0.000	0.000
Class 4	0.000	0.000	0.000	0.000	0.000
Class 5	0.231	0.000	1.000	0.375	0.469
Class 6	0.000	0.000	0.000	0.000	0.000
Class 7	0.759	0.043	0.733	0.746	0.705
Weighted-average	0.584	0.202	0.536	0.540	0.384

Table 47 Classification performance w.r.t different evaluation measures for Glass data using PART

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	0.757	0.174	0.679	0.716	0.569
Class 2	0.658	0.152	0.704	0.680	0.514
Class 3	0.176	0.071	0.176	0.176	0.105
Class 4	0.000	0.000	0.000	0.000	0.000
Class 5	0.615	0.015	0.727	0.667	0.650
Class 6	0.889	0.005	0.889	0.889	0.884
Class 7	0.828	0.022	0.857	0.842	0.818
Weighted-average	0.682	0.121	0.684	0.682	0.808

Table 48 Classification performance w.r.t different evaluation measures for Glass data using ZeroR

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	0.000	0.000	0.000	0.000	0.000
Class 2	1.000	1.000	0.355	0.524	0.000
Class 3	0.000	0.000	0.000	0.000	0.000
Class 4	0.000	0.000	0.000	0.000	0.000
Class 5	0.000	0.000	0.000	0.000	0.000
Class 6	0.000	0.000	0.000	0.000	0.000
Class 7	0.000	0.000	0.000	0.000	0.000
Weighted-average	0.355	0.355	0.126	0.186	0.000

Table 49 Classification performance w.r.t different evaluation measures for Glass data using NSGA-II

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	0.814	0.208	0.655	0.726	0.579
Class 2	.658	0.174	0.676	0.667	0.487
Class 3	0.000	0.010	0.000	0.000	− 0.029
Class 4	0.000	0.000	0.000	0.000	0.000
Class 5	0.846	0.020	0.733	0.786	0.773
Class 6	0.556	0.024	0.500	0.526	0.505
Class 7	0.793	0.016	0.885	0.836	0.814
Weighted-average	0.682	0.135	0.640	0.657	0.538

Wine Dataset It is impractical to generate candidate rules by examining all the $(5+1)^{13}$ combinations of antecedent linguistic values. Table 7 shows the comparative performance.

Table 50 Classification performance w.r.t different evaluation measures for Glass data using RBFN + NSGA-II

Class	TP Rate	FP Rate	Precision	*F*-Measure	MCC
Class 1	0.800	0.215	0.644	0.713	0.558
Class 2	0.645	0.138	0.721	0.681	0.521
Class 3	0.118	0.025	0.286	0.167	0.140
Class 4	0.000	0.000	0.000	0.000	0.000
Class 5	0.692	0.015	0.750	0.720	0.703
Class 6	0.778	0.005	0.875	0.824	0.818
Class 7	0.897	0.032	0.813	0.852	0.829
Weighted-average	0.696	0.127	0.682	0.682	0.568

Glass Dataset The glass dataset is a difficult classification problem with large overlaps between different classes in the pattern space. So it may be difficult to design a compact fuzzy rule-based classification with high accuracy by a small number of short fuzzy rules. Table 8 shows the result of our method.

The number of non-dominated set versus classification rate of both methods for the datasets IRIS, WINE, and GLASS datasets is shown in Figs. 3, 4, and 5, respectively.

5.3.2 *Comparative Analysis with Crisp Rule Mining Approaches*

Here, we illustrate the classification performance of our model through various evaluation measures derived from confusion matrix and then compare with five crisp rule mining approaches like Decision Table, One Rule (OneR), PART, JRip, and Zero Rule (ZeroR).

For the case of Iris dataset, the distinct confusion matrices obtained with respect to different methods are illustrated through Tables 9, 10, 11, 12, 13, 14, and 15. Similarly, the different measures like TP Rate, FP Rate, Precision, *F*-Measure, and MCC are illustrated through Tables 16, 17, 18, 19, 20, 21, and 22. For the case of Wine dataset, the distinct confusion matrices obtained with respect to different methods are illustrated through Tables 23, 24, 25, 26, 27, 28, and 29. Similarly, the different measures like TP Rate, FP Rate, Precision, *F*-Measure, and MCC are illustrated through Tables 30, 31, 32, 33, 34, 35 and 36. For the case of Glass dataset, the distinct confusion matrices obtained with respect to different methods are illustrated through Tables 37, 38, 39, 40, 41, 42, and 43. Similarly, the different measures like TP Rate, FP Rate, Precision, *F*-Measure, and MCC are illustrated through Tables 44, 45, 46, 47, 48, 49, and 50.

6 Conclusions

In this paper, we have described a surrogate-assisted NSGA-II model by RBFNs for fuzzy classification rule mining problem. Emphasis was given on the comprehensibility of fuzzy rule-based system rather than only focusing on classification performance. The proposed algorithm was evaluated on three benchmark datasets with encouraging results. The simulation studies show that a small number of linguistically interpretable fuzzy rules were found by the proposed algorithm for designing comprehensible fuzzy rule-based classification system for high-dimensional data classification problem.

Acknowledgements This work was partly supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (R0124-16-0002), Emotional Intelligence Technology to Infer Human Emotion and Carry on Dialogue Accordingly.

References

1. Witten, I.H., Frank, E., Hall, M.A., Pal, C.: Data Mining: Practical Machine Learning Tools and Techniques, 4th edn. Morgan Kaufmann, Burlington (2016)
2. Weiss, S.M., Kapouleas, I.: An empirical comparison of pattern recognition, neural nets and machine learning classification methods. In: Readings in Machine Learning, pp. 177–183. Morgan Kaufmann Publishers Inc., Burlington (1990)
3. Lim, T.S., Loh, W.Y., Shih, Y.S.: A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Mach. Learn. **40**(3), 203–228 (2000)
4. Freitas, A.A.: A survey of evolutionary algorithms for data mining and knowledge discovery. In: Ghosh, A., Tsutsui, S. (eds.) Advances in Evolutionary Computation. Springer, Berlin (2001)
5. Smith, E., Dike, B.A., Stegmann, S.A.: Fitness inheritance in genetic algorithms. In: Proceedings of the 1995 ACM Symposium on Applied Computing, pp. 345–350. ACM Press, New York (1995)
6. Ishibuchi, H., Nakasima, T.: Linguistic rule extraction by genetic-based machine learning. In: Proceedings of Genetic and Evolutionary Computation Conference, pp. 195–202. Morgan Kaufmann (2000)
7. Ishibuchi, H., Murata, T., Turksen, I.B.: Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. Fuzzy Sets Syst. **89**(2), 135–150 (1997)
8. Zimmermann, H.J.: Fuzzy Set Theory and Its Applications. Springer, Berlin (2011)
9. Kramer, O.: Genetic Algorithm Essentials. Studies in Computational Intelligence, vol. 679. Springer, Cham (2017)
10. Fernandez, A., Garcia, S., Luengo, J., Bernado-Mansilla, E., Herrera, F.: Genetics-based machine learning for rule induction: state of art, taxonomy and comparative study. IEEE Trans. Evolut. Comput. **14**(6), 913–943 (2010)
11. Herrera, F.: Genetic fuzzy systems: taxonomy, current research trends and prospects. Evolut. Intell. **1**(1), 27–46 (2008)
12. Smith, S.F.: A learning system based on genetic algorithms. Ph.D. dissertation, University of Pittsburg, Pittsburg PA (1980)
13. Ong, Y.S., Nair, P.B., Keane, A.J.: Evolutionary optimization of computationally expensive problems via surrogate modelling. Am. Inst. Aeronaut. Astronaut. J. **41**(4), 687–696 (2003)
14. Jin, Y., Olhofer, M., Sendhoff, B.: A framework for evolutionary optimization with approximate fitness functions. IEEE Trans. Evolut. Comput. **6**(5), 481–494 (2002)
15. Shi, L., Rasheed, K.: A survey of fitness approximation methods applied in evolutionary algorithms. In: Goh, C.K., Tenne, Y. (eds.) Computational Intelligence in Expensive Optimization Problem, pp. 3–28. Springer, Berlin (2010)
16. Eby, D., Averill, R., Panch, W., Goohman, E .: Evaluation of injection island model GA performance on flywheel design optimization. In: Third Conference on Adaptive Computing in Design and Manufacturing, pp. 121–136. Springer (1998)
17. Sefrioui, M., Periaux, J.: A hierarchical genetic algorithm using multiple models for optimization. In: Parallel Problem Solving From Nature, PPSN VI, pp. 879-888. Springer, Berlin (2000)
18. Abbond, K., Schoenauer, M.: Surrogate Deterministic Mutation. Artificial Evolution, vol. 2310, pp. 103–115. Springer, Berlin (2002)
19. Rasheed, K.: Informed operators: speeding up genetic-algorithm-based design optimization using reduced models. In: Proceedings of Genetic and Evolutionary Computation Conference, pp. 628–635. Morgan Kaufmann, Las Vegas (2000)
20. Papadrakakis, M., Lagaros, N., Tsompanakis, Y.: Optimization of large-scale 3D trusses using evolutionary strategies and neural networks. Int. J. Space Struct. **14**(3), 211–223 (1997)
21. Emmerich, M., Giotis, A., Ozdenir, M., Back, T., Giannakoglou, K.: Meta-model-assisted evolution strategies. In: International Conference on parallel problem solving from nature, pp. 361–370. Springer, Berlin (2002)
22. Jin, Y.: Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement. IEEE Trans. Fuzzy Syst. **8**, 212–222 (2000)
23. Ishibuchi, H., Nakashima, T., Morisawa, T.: Voting in fuzzy rule-based systems for pattern classification problems. Fuzzy Sets Syst. **103**, 223–238 (1999)
24. Ishibuchi, H., Nozaki, K., Yamamoto, N., Tanaka, H.: Selecting fuzzy if-then rules for classification problem using genetic algorithms. IEEE Trans. Fuzzy Syzt. **3**(3), 260–270 (1995)
25. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evolut. Comput. **6**(2), 182–197 (2002)
26. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the performance of the strength of Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Communication Networks Lab, Swiss Federal Institute of Technology, Zurich (2001)
27. Watanabe, S., Hiroyassu, T., Miki, M.: Multi-objective rectangular packing problem and its applications. In: Proceedings of 2nd International Conference on Multi-criterion Optimization, pp. 565–577 (2003)
28. Ghosh, A., Das, M.K.: Non-dominated rank based sorting genetic algorithms. Fundam. Inf. **83**(3), 231–252 (2008)
29. Sanchez, L., Otero, J., Couso, I.: Obtaining linguistic fuzzy rule-based regression models from imprecise data with multiobjective genetic algorithms. Soft Comput. **13**(5), 467–479 (2009)
30. Ishibuchi, H., Kuwajima, I., Nojima, Y.: Use of Pareto-optimal and near Pareto-optimal candidate rules in genetic fuzzy rule selection. In: Melin, P., et al. (eds.) Analysis and Design of Intelligent Systems Using Soft Computing Techniques, pp. 387–396. Springer, Berlin (2007)
31. Ji-lin, C., Yuan-long, H., Zong-yi, X., Li-min, J., Zhong-zhi, T.: A multi-objective genetic-based method for design fuzzy classification systems. Int. J. Comput. Sci. Netw. Secur. **6**(8A), 110–117 (2006)

32. Zhang, Y., Wu, X.B., Xing, Z.Y., Hu, W.L.: On generating interpretable and precise fuzzy systems based on Pareto multi-objective cooperative co-evolutionary algorithm. Appl. Soft Comput. **11**(1), 1284–1294 (2011)
33. Nojima, Y., Kaisho, Y., Ishibuchi, H.: Accuracy improvement of genetic fuzzy rule selection with candidate rule addition and membership tuning. In: Proceedings of IEEE International Conference on Fuzzy Systems, Barcelona, pp. 1–8 (2010)
34. Alcalá, R., Nojima, Y., Herrera, F., Ishibuchi, H.: Generating single granularity-based fuzzy classification rules for multi-objective genetic fuzzy rule selection. In: Proceedings of IEEE International Conference on Fuzzy Systems, pp. 1718–1723 (2009)
35. Nojima, Y., Ishibuchi, H.: Incorporation of user preference into multi-objective genetic fuzzy rule selection for pattern classification problems. Artif. Life Robot. **14**(3), 418–421 (2009)
36. Ishibuchi, H., Nakashima, Y., Nojima, Y.: Performance evaluation of evolutionary multiobjective optimization algorithms for multi-objective fuzzy genetics-based machine learning. Soft Comput. **15**(12), 2415–2434 (2011)
37. Alcalá, R., Nojima, Y., Herrera, F., Ishibuchi, H.: Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions. Soft Comput. **15**(12), 2303–2318 (2011)
38. Trawiski, K., Cordon, O., Quirin, A.: A study on the use of multiobjective genetic algorithms for classifier selection in FURIA-based fuzzy multiclassifiers. Int. J. Comput. Intell. Syst. **5**(2), 231–253 (2012)
39. Fazzolari, M., Giglio, B., Alcala, R., Marcelloni, F., Herrera, F.: A study on the application of instance selection techniques in genetic fuzzy rule-based classification systems: accuracy-complexity trade-off. Knowl. Based Syst. **54**, 32–41 (2013)
40. Fazzolari, M., Alcala, R., Herrera, F.: A multi-objective evolutionary method for learning granularities based on fuzzy discretization to improve the accuracy-complexity trade-off of fuzzy rule-based classification systems: D-MOFARC algorithm. Appl. Soft Comput. **24**, 470–481 (2014)
41. Antonelli, M., Ducange, P., Marcelloni, F.: A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers. Inf. Sci. **283**(1), 36–54 (2014)
42. Gorzaczany, M.B., Filip, R.: A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability. Appl. Soft Comput. **40**, 206–220 (2016)
43. Antonelli, M., Ducange, P., Lazzerini, B., Marcelloni, F.: Multi-objective evolutionary design of granular rule-based classifiers. Granul. Comput. **1**(1), 37–58 (2016)
44. Rudziski, F.: A multi-objective genetic optimization of interpretability-oriented fuzzy rule-based classifiers. Appl. Soft Comput. **38**, 118–133 (2016)
45. Antonelli, M., Ducange, P., Lazzerini, B., Marcelloni, F.: Learning knowledge bases of multi-objective evolutionary fuzzy systems by simultaneously optimizing accuracy, complexity and partition integrity. Soft Comput. **15**(12), 2335–2354 (2011)
46. Gacto, M.J., Alcala, R., Herrera, H.: A multi-objective evolutionary algorithm for tuning fuzzy rule-based systems for preserving interpretability. In: Proceedings of the Joint International Fuzzy Systems Association World Congress and the European Society for Fuzzy Logic and Technology Conference (IFSA/EUSFLAT 2009), pp. 1146–1151 (2009)

Harihar Kalia is an Assistant Professor in the Department of Computer Science and Engineering, Seemanta Engineering College, Jharpokharia, Mayurbhanj, Odisha. He received his M.Sc. degree in Mathematics from Ravenshaw University, Odisha, in 1995, and the M.Tech. degree in Computer Science from Utkal University, Vani Vihar, Odisha, in 2002. He completed his Ph.D. work in the area of multi-objective fuzzy rule mining in Department Information and Communication Technology, Fakir Mohan University, Odisha, India, in 2015. His area of interest includes: data mining, multi-objective optimization, fuzzy logic, evolutionary algorithms, and hybrid systems. He has about 19-year teaching experience, in teaching both graduate and postgraduate students. He is a member of Editorial and Reviewer Board in International Journal of Rough Computing and Intelligent Systems (IJRCIS), Reviewer in International Journal of Uncertainty, Fuzziness and Knowledge Based Systems (World Scientific), Editorial Review Board Member in International Journal of Applied Meta heuristic Computing (IJAMC), Reviewer in International Journal of Ambient Computing and Intelligence (IJACI), IGI Global, Reviewer in Knowledge Based System, Elsevier, and Reviewer in International Journal of Operational Research, Inderscience.

Satchidananda Dehuri is an Associate Professor in the Department of Information and Communication Technology, Fakir Mohan University, Balasore, Odisha, since 2010. He received his Ph.D. in Computer Science from the Utkal University, Odisha, in 2006. He was a BOYSCAST Fellow in the Soft Computing Laboratory, Yonsei University, Seoul, Korea, under the BOYSCAST Fellowship Program of DST, Government of India. In 2010, he received the Young Scientist Award in Engineering and Technology for the year 2008 from the Odisha Vigyan Academy, Department of Science and Technology, Government of Odisha. His research interests include soft computing and data mining.

Ashish Ghosh is a Professor in Machine Intelligence Unit, Indian Statistical Institute. He has already published more than 150 research papers in internationally reputed journals and refereed conferences and has edited eight books. His current research interests include pattern recognition and machine learning, data mining, image analysis, remotely sensed image analysis, video image analysis, soft computing, fuzzy sets and uncertainty analysis, neural networks, evolutionary computation, and bioinformatics. He received the prestigious and most coveted Young Scientists Award in Engineering Sciences from the Indian National Science Academy in 1995 and in Computer Science from the Indian Science Congress Association in 1992.

Sung-Bae Cho received the Ph.D. degrees in computer science from KAIST (Korea Advanced Institute of Science and Technology), Taejeon, Korea, in 1993. He was an Invited Researcher of Human Information Processing Research Laboratories at ATR (Advanced Telecommunications Research) Institute, Kyoto, Japan from 1993 to 1995, and a Visiting Scholar at University of New South Wales, Canberra, Australia in 1998. He was also a Visiting Professor at University of British Columbia, Vancouver, Canada from 2005 to 2006. Since 1995, he has been a Professor in the Department of Computer Science, Yonsei University. His research interests include neural networks, pattern recognition, intelligent man-machine interfaces, evolutionary computation, and artificial life. He is a Senior Member of IEEE and a Member of the Korea Information Science Society, the IEEE Computer Society, the IEEE Systems, Man, and Cybernetics Society, and the Computational Intelligence Society.