Induction of accurate and interpretable fuzzy rules from preliminary crisp representation

Chen, Tianhua; Shang, Changjing; Su, Pan; Shen, Qiang

Published in:
Knowledge-Based Systems

DOI:
10.1016/j.knosys.2018.02.003

Publication date:
2018

Citation for published version (APA):
Chen, T., Shang, C., Su, P., & Shen, Q. (2018). Induction of accurate and interpretable fuzzy rules from preliminary crisp representation. *Knowledge-Based Systems*, *146*, 152-166. https://doi.org/10.1016/j.knosys.2018.02.003

Knowledge-Based Systems 146 (2018) 152–166

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Induction of accurate and interpretable fuzzy rules from preliminary crisp representation

Tianhua Chen[a], Changjing Shang[b], Pan Su[c], Qiang Shen[b,*]

[a] Department of Computer Science, School of Computing and Engineering, University of Huddersfield, Huddersfield, UK
[b] Department of Computer Science, Institute of Mathematics, Physics and Computer Science, Aberystwyth University, Aberystwyth, UK
[c] School of Control and Computer Engineering, North China Electric Power University, Baoding, China

ARTICLE INFO

Article history:
Received 2 November 2017
Revised 11 January 2018
Accepted 2 February 2018
Available online 9 February 2018

Keywords:
Fuzzy rule-based systems
Interpretable fuzzy rules
Fuzzy rule learning
Crisp rules
Fuzzy rule-based classification

ABSTRACT

This paper proposes a novel approach for building transparent knowledge-based systems by generating accurate and interpretable fuzzy rules. The learning mechanism reported here induces fuzzy rules via making use of only predefined fuzzy labels that reflect prescribed notations and domain expertise, thereby ensuring transparency in the knowledge model adopted for problem solving. It works by mapping every coarsely learned crisp production rule in the knowledge base onto a set of potentially useful fuzzy rules, which serves as an initial step towards an intuitive technique for similarity-based rule generalisation. This is followed by a procedure that locally selects a compact subset of the emerging fuzzy rules, so that the resulting subset collectively generalises the underlying original crisp rule. The outcome of this local procedure forms the input to a global genetic search process, which seeks for a trade-off between accuracy and complexity of the eventually induced fuzzy rule base while maintaining transparency. Systematic experimental results are provided to demonstrate that the induced fuzzy knowledge base is of high performance and interpretability.

1. Introduction

Knowledge-based systems (KBSs) aim to represent knowledge explicitly via tools such as production or if-then rules, which allow such a system to reason about how it reaches a conclusion and to provide explanation of its reasoning to the user [1]. Fuzzy systems have been considered effective in building KBSs [2,3], particularly in environments where the information (in terms of data or knowledge) is imprecise in nature. Fuzzy KBSs are able to deal with vague concepts that are fundamental to natural languages in practical reasoning and decision making. This facilitates the combination of the information obtained from physical sensory measurements and that from experts' descriptions directly using a natural language, which in turn flexibly supports the design and implementation of such KBSs in effectively addressing real-world problems. Many approaches [4–8] have been proposed for generating and learning fuzzy KBSs to represent the input–output behaviour of a certain problem, including the development of fuzzy rule-based classification systems (FRBCSs) where the output of a learned system is typically crisp and discrete.

One of the most important advantages of fuzzy systems lies in their inherent interpretability as they support the explicit formulation of, and inference with, domain knowledge, gaining insights into the complex problems and facilitating the explanation of their solutions. However, unlike criteria such as accuracy that can be used to precisely and objectively measure how good a fuzzy model is with respect to the real system being modelled, interpretability is a subjective property, which largely depends on the person who makes the assessment. Due to the subjectivity nature, interpretability may be affected by a range of practical issues, especially regarding the representation of the underlying concepts and knowledge in the problem domain. Different approaches [9–13] have been proposed to study interpretability within the general area of fuzzy systems. Although there lacks a commonly accepted mechanism to adjudge interpretability, complexity-based and semantics-based methods are typically considered when designing a fuzzy KBS. Complexity-based interpretability aims to reduce the complexity of a fuzzy model in terms of the number of rules and the number of variables or their labels per rule. Semantics-based interpretability aims to preserve the semantics of the membership functions (MFs), such that the fuzzy rules make use of meaningful linguistic labels.

The incorporation of intuitive expert knowledge into linguistic rules through the use of predefined fuzzy sets is desirable to

* Corresponding author.
E-mail address: qqs@aber.ac.uk (Q. Shen).

https://doi.org/10.1016/j.knosys.2018.02.003
0950-7051/

effectively interpret a fuzzy model. This allows for enhanced transparency in both the learned models themselves and the inferences performed by running the learned models [12,14]. For many real-world applications (e.g., medical diagnosis [15,16] and intelligence data analysis [17,18]), the use of a fixed and predefined quantity space per variable is indeed a must. Thus, conventional approaches to learning fuzzy KBS models tend to subsequently impose semantic constraints over MFs in an effort to boost model accuracy by modifying the definition of the fuzzy sets [11]. This may help improve the accuracy of the resulting learned model, but such computation may adversely affect the exactly prescribed meaning of the given labels. This in turn may destroy the interpretability of the overall rule model that employs such changed linguistic labels.

Generally agreed knowledge from a certain problem domain should be retained and built into the design of a KBS, be it fuzzy or not. For FRBCSs, the labelled fuzzy terms only make sense if their underlying definitions are consistent with the common notations that the users understand. This implies that in developing FRBSCs, domain expertise in terms of fuzzy rules that utilise predefined fuzzy sets should be maintained if possible. Direct use of domain expertise also makes it easier for experts to verify the results returned by a fuzzy KBS, forming a sharp contrast with black-box systems such as neural networks [19] that can achieve high performance, but their solutions are difficult to comprehend. Thus, the induction of a fuzzy rule base should be independent of the acquisition of the data base that specifies the definitions of fuzzy sets for the variables (which can be assumed to be given). As such, the conventional approach that induces a rule base by iteratively refining a database, using the information extracted from the future iteration to alter the definitions of the currently learned rules will not work as the specification of the underlying fuzzy terms may be badly distorted if not completely destroyed. This observation has inspired the research reported herein, which entails the automatic generation of accurate and interpretable fuzzy classification rules, where the use of fixed and predefined quantity space is presumed to retain for semantic interpretability.

The present work therefore, promotes an alternative approach, where a fuzzy model is initialised by utilising preliminary existing crisp rules that have been generated by a certain crisp rule-based learning mechanism. A similarity-based mapping is then performed over the rule base, mapping each existing crisp rule onto a set of potentially useful fuzzy rules that only use predefined fuzzy sets. This approach follows on the intuitive presumption that each of the given crisp rules points to a certain place in the search space where desirable fuzzy rules potentially exist. Predefined fuzzy sets reflecting domain-specific knowledge remain unchanged throughout the modelling and inference processes, ensuring semantic interpretability. To balance between accuracy and complexity of the final rule base, a trade-off is made by a procedure that locally selects a compact subset of the potential fuzzy rules per crisp rule, guaranteeing that each resulting subset collectively generalises the corresponding original crisp rule. The outcome of this local selection forms the input to a global genetic rule selection process, leading to a required final interpretable and accurate fuzzy model.

To demonstrate the generalisation capability and versatility of the proposed approach, two crisp rule-based classifiers following distinct rule induction strategies are utilised to initialise the proposed work. Systematic experiments compare the present work against both popular fuzzy rule-based and non-fuzzy-rule-based learning classifiers using 16 benchmark data sets. The experimental investigations also include analyses of the complexity of the learned model and that of the effect of local rule selection procedure in relation to functional generalisation.

The reminder of this paper is organised as follows. Section 2 introduces the background of fuzzy rules and fuzzy classification systems. Section 3 describes the proposed methodology of generating an interpretable fuzzy rule base by utilising a given crisp rule set and predefined fuzzy labels. Section 4 presents and discusses comparative experimental results. Section 5 concludes the paper and outlines ideas for further development.

2. Background

This section briefly reviews the representation of fuzzy rule-based models for the development of FRBCSs and further motivates the development of the present work.

2.1. Representation of fuzzy rule models

The task of learning an FRBCS is to find a finite set of fuzzy production or if-then rules capable of classifying a given input. Without losing generality, the classification system to be modelled is herein assumed to be multiple-input-single-output, receiving n-dimensional input patterns and producing one output which is determined to be of one of the M classes. The fuzzy rule set to be induced is required to perform the mapping φ: $X^n \rightarrow Y$, where $X^n = X_1 \times X_2 \times \cdots \times X_n$, $X_1, X_2, \ldots, X_n$ are the domains of discourse of the input variables and Y represents the set of possible output classes of a cardinality of M. The information about the behaviour of the system is described by a set of input-output example pairs E, where for each (cross-product) instantiation of the input variables $\bar{x}^p = (x_1^p, x_2^p, \ldots, x_n^p)^T, x_i^p \in X_i, i = 1, 2, \ldots, n$, an associated class $y^p \in Y$ is indicated.

In general, a fuzzy if-then rule F_j can be represented as follows:

$$\text{If } x_1 \text{ is } D_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } D_{jn}, \text{ Then class is } y^{F_j} \tag{1}$$

where $j = 1, 2, \ldots, N$, with N denoting the number of all such fuzzy rules within the system; $x_i, i = 1, \ldots, n$ are the underlying domain variables, jointly defining the n-dimensional pattern space and respectively taking values from X_i; $D_{ji} \in X_i$ denotes a fuzzy set that the variable x_i may take; and $y^{F_j} \in Y$ is the consequent of the fuzzy rule F_j that is to be assigned to one of the M possible output classes.

Without complicating the representation scheme fuzzy rules adopted in this paper do not involve the use of rule weights. Their involvement could further improve classifier performance, but may pay the price of affecting semantic transparency, as rule weights change the normality of antecedent fuzzy sets [20]. Including rule weights will also increase computational effort. Unless otherwise stated, in this work, each D_{ji} in the above description is a semantic fuzzy set for the variable x_i, which is predefined and fixed throughout both the modelling and inference processes.

The notion of compatibility matching degree of each training point $\bar{x}^p$, with respect to the rule $F_j, j = 1, 2, \ldots, N$, is defined within the n-dimensional fuzzy subspace $D_{j1} \times D_{j2} \times \cdots \times D_{jn}$, such that

$$\mu_{F_j}(\bar{x}^p) = \prod_{i=1}^{n} \mu_{D_{ji}}(x_i^p) \tag{2}$$

where $\mu_{D_{ji}}(x_i^p)$ represents the matching degree regarding the ith corresponding antecedent x_i. A void in D_{ji} indicates that no term from the domain of the variable x_i is present in F_j, implying that the variable is irrelevant to making a certain classification on the point $\bar{x}^p$. To determine the class label of a newly presented pattern with a set of fuzzy rules, the popular single-winner-taking-all strategy is adopted for interpretation purpose, such that the point is identified with the class label that is of the following maximum matching degree from the available rule base:

$$y^p = y^{F_j} \,|\, \arg\max_{j=1,2,\ldots,N} \mu_{F_j}(x^p) \tag{3}$$

If two or more classes take the same maximum value or the total compatibility degree is zero, no pattern can be uniquely classified. To force a classification (if desired), such a pattern may be assigned with a default class label that is associated with most training instances.

2.2. Motivational observations

Depending on how predefined membership functions of the underlying fuzzy sets are generated, this paper categories them into three cases:

1. Fuzzy sets that are defined through consulting domain experts. This is preferred given the original motivation of incorporating human knowledge into the system design. However, it will have to take extra work consulting with domain experts, who may not be available nor always consistent in providing such information.
2. Fuzzy sets that are uniformly partitioned in the universe of discourse, assuming that the underlying problem domain involves data that is uniformly distributed. In case of no domain expertise available, membership functions could be built this way. This has been considered being interpretable given limited domain knowledge [11] and is therefore, also investigated in this paper. Many interpretability indices have been constructed by measuring the differences between the fuzzy sets specified in this manner and the fuzzy sets that are computationally optimised. Obviously, a key disadvantage of this approach is that the fuzzy sets may not reflect the true distribution of the underlying data, which would result in great performance loss.
3. Fuzzy sets that are defined with respect to an analysis of the characteristics of the underlying data, without direct involvement of domain expertise. That is, partitioning the variable domains through clustering subject to the constraint that is similar to the second approach above, by discretising each feature space into a certain number of fuzzy sets. However, depending on how the training instances are sampled from the domain, this may not necessarily reflect the real characteristics of the real problem and may result in counter-intuitive definitions, thereby misleading the interpretability of the learned fuzzy rule model using such defined fuzzy sets.

In the literature, a variety of techniques have been proposed to extract a set of linguistic fuzzy rules with fixed (and often, uniformly divided) fuzzy sets. To counter against the stochastic nature of many real-world problems, evolutionary computation approaches are often taken in developing such techniques. For instance, FH-GBML is a hybrid fuzzy genetics algorithm [21,22]. It uses the Pittsburgh style to encode a set of fuzzy rules as an individual, while using the Michigan style for partially modifying each rule set as a heuristic for mutation. The popular SLAVE2 method [23,24] learns rules of a disjunctive normal form through an iterative algorithm that is implemented with a GA, where each chromosome represents a single rule. A pattern-tree learning classifier (dubbed PTTD) is introduced in [7,25], depicted in a hierarchical, tree-like structure, whose inner nodes are marked with generalised logical operators and leaf nodes associated with fuzzy predicates on the inputs. GP-COACH [26] is a genetic programming-based cooperative-competitive learning approach, which also learns rules of a disjunctive normal form with a coding scheme that expresses one rule per tree. SGERD proposes a steady-state GA to extract a compact set of fuzzy rules by exploiting specific rule and data dependent parameters [27].

Typically, these approaches make consistent use of uniformly divided MFs that support interpretability from the view point of maintaining model semantics. However, this usually suffers from the curse of dimensionality as the number of inputs increases. Besides, many of the rules generated may not cover any training pattern at all. The subset of interesting rules that cover certain training data may be rather small as compared to the total; much effort of the search may have been wasted in order to find that small subset. Therefore, instead of considering all of the possible combinations of the input and class variables, it is herein proposed to initially utilise existing crisp rule generation techniques that are able to efficiently generate sufficiently effective rules while focusing on given data, without resorting to pure and brute force search. Being fundamentally data-driven, such a rule generation method will omit the empty parts of the input space, substantially expediting the overall learning process.

Each of the generated crisp rules forms a certain partition of the entire problem space, and points to those parts in which desirable fuzzy rules may potentially exist. Each crisp rule is then locally mapped onto a compact set of interpretable fuzzy rules involving only predefined meaningful fuzzy labels. This is followed by a global genetic rule generalisation and selection procedure to produce a fuzzy model that is of high performance and interpretability (in both model semantics and model complexity). This proposed approach is different from what is often done when utilising crisp rule-based classifiers to initialise potential fuzzy classifiers, either through variable or feature selection techniques [28], or by fitting and fine tuning the generated crisp intervals into certain parameterised MFs [29] (which would of course result in semantic loss).

3. Generating interpretable fuzzy classifiers from crisp rules

This section details the proposed approach. In particular, Section 3.1 presents a heuristic method for mapping preliminary crisp rules onto potential fuzzy rules, while involving only predefined meaningful fuzzy labels. Section 3.2 describes a procedure that locally selects a compact subset of the potential fuzzy rules for each original crisp rule, by ensuring that the selected rules collectively generalise the original. Section 3.3 introduces a global search mechanism (implemented with a GA) for the acquisition of an interpretable fuzzy rule base with a tradeoff between accuracy and complexity. Section 3.4 summarises the proposed approach together with a complexity analysis.

3.1. Mapping crisp rules to fuzzy rules

3.1.1. Heuristic mapping

To generate an accurate and compact set of interpretable fuzzy rules effectively and efficiently, it is useful to have an initial focus on where the potentially meaningful rules may reside without going through an exhaustive search. An easily conceived way to implement this is to make use of an initial set of if-then crisp rules available (e.g., generated by a certain learning mechanism or provided by domain experts), even though such rules might not be very accurate. Without losing generality, suppose that a crisp rule $C_j, j = 1, 2, \ldots, N$ (with N denoting the number of all crisp rules available) is given as follows:

$$\text{If } x_1 \text{ is } I_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } I_{jn}, \text{ Then class is } y^{C_j} \tag{4}$$

where $x_1, x_2, \ldots, x_n$ represent the underlying domain variables, jointly defining an n-dimensional input pattern space; I_{ji}, $i \in \{1, 2, \ldots, n\}$, is the crisp interval of the antecedent variable x_i; and y^{C_j} is a class label, acting as the rule consequent (which may be encoded as an integer for simplicity in implementation).

In order to approximate the modelling problem with a set of fuzzy rules as of Eq. (1), where variables are described with predefined fuzzy sets instead of crisp intervals, a procedure is required to convert crisp intervals into the corresponding fuzzy terms. The idea to implement such a mapping is to use a similarity measure

between a crisp interval and each of the predefined fuzzy sets describing the same variable, such that only those fuzzy sets are considered valid whose similarity values are above a user-defined threshold or confidence level η.

A heuristic is employed herein to obtain the set of potentially useful interpretable rules by mimicking the method of [14]. This heuristic procedure can be summarised as follows, with an example given to further explain how it works in the following subsection. It first builds up a layered graph, where a node in a certain layer contains a number of predefined fuzzy sets in association with each existing crisp interval per variable (each of which has a similarity measure with the original crisp interval above η). This process iterates until all the corresponding crisp sets that are associated with all the nodes within each layer have been successfully replaced by predefined fuzzy sets. A path from one layer to another can be built by connecting one and only one node from each layer. As such, each resultant path can be interpreted as a possible interpretable fuzzy rule which coarsely approximates the given crisp rule under mapping.

Note that crisp intervals in a crisp rule are themselves crisp sets, each of which can be seen as a special case of fuzzy sets. Thus, the similarity between a crisp set and a fuzzy set can be generalised as the similarity between two fuzzy sets. There are many such similarity metrics available in the literature. The following set-theoretic based similarity measure is adopted in this work (owing to its popularity though others may be used as an alternative):

$$S(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|} \tag{5}$$

where A and B denote two fuzzy sets; $|.|$ represents the cardinality of a fuzzy set; and $\bigcap$ and $\bigcup$ denote set intersection and union, respectively.

From the above, the similarity between a predefined fuzzy set D_{ji} and a crisp set I_{ji} regarding the i-th variable within a given rule C_j can be rewritten as:

$$S(D_{ji}, I_{ji}) = \frac{\sum_{\bar{x}^p \in E_j^i}[\mu_{D_{ji}}(x_i^p) \wedge \mu_{I_{ji}}(x_i^p)]}{\sum_{\bar{x}^p \in E_j^i}[\mu_{D_{ji}}(x_i^p) \vee \mu_{I_{ji}}(x_i^p)]} \tag{6}$$

where $\wedge$ and $\vee$ represent the minimum and maximum operator, respectively, and

$$E_j^i = \left\{ \bar{x}^p \mid \prod_{k \neq i} \mu_{I_k}(x_k^p) > 0, \bar{x}^p \in E_{\text{trn}} \right\} \tag{7}$$

where $\bar{x}^p$ stands for an instance from the training data set E_{trn}; and the check of $\mu_{I_k}(x_k^p) > 0$ is to ensure that the training instance intersects with all antecedent variables, except the i-th variable itself.

The computation effort required for calculating this similarity measure is significantly lighter than what it may appear at the first sight. This is because in general, the set of training instances used for calculating the similarity is not the entire training set, but the subset of training data specified by Eq. (7). However, it does not necessarily ensure a good coverage of the original crisp rule unless the threshold value is set very low. Yet, a low threshold implies many matching nodes to be retained and hence, many potential fuzzy rules to be created. A large number of rules not only increases computational complexity but also deteriorates the interpretability of the learned model. A way to reduce the impact of this sensitivity in parameter setting is to introduce another user-defined parameter T such that a very low threshold value may be set, but only those T most similar fuzzy sets may be retained per variable.

3.1.2. Illustrative example

To illustrate the basic idea of the above heuristic process, consider a crisp rule C under mapping as follows:

$$\text{If } x_1 \text{ is } I_1 \text{ and } x_2 \text{ is } I_2, \text{ Then class is } y^C \tag{8}$$

where I_1 and I_2 are two crisp sets describing the two input variables x_1 and x_2, respectively. Suppose that a collection of predefined fuzzy sets $\{D_{ji} | j = 1, 2, \ldots, k_i\}$ per variable $(x_i, i = 1, 2)$ is provided. For simplicity, let $k_i = 3, i = 1, 2$. In particular, the three semantic fuzzy sets are defined for each variable such that x_1 may take a value on either of $D_{11} =$ low, $D_{21} =$ medium, $D_{31} =$ high, and x_2 on either of $D_{12} =$ small, $D_{22} =$ medium, $D_{32} =$ large.

Following the above-introduced heuristic procedure, the first layer of the hierarchical graph to be built is set to work on the crisp set of the first antecedent variable first, i.e., I_1 in this case (assuming the strategy of first come first service). Then, a node is created for each of the predefined corresponding fuzzy sets $D_{j1}, j = 1, 2, 3$ if it has a similarity value greater than a given threshold η (which is here set to 0 by default) to I_1. Suppose that $S(I_1, D_{11}) = 0,\ S(I_1, D_{21}) = 0.75,$ and $S(I_1, D_{31}) = 0.3$. With the default threshold, the nodes representing the two valid fuzzy sets of D_{21} and D_{31} are retained in the graph. The similar process is repeated for the next antecedent variable. From which, all retained nodes in a preceding layer are connected to those in the immediate subsequent layer. The result of this mapping process for the example is shown in Fig. 1.

Once such a graph is generated, each path becomes an emerging fuzzy rule, with the antecedent variables described by corresponding fuzzy sets, while the rule consequent remains to be the same as that of the original crisp rule. This leads to a set of possible fuzzy rules involving the use of only predefined fuzzy sets. For the example, the resultant rules are:

Rule F_1: If x_1 is medium and x_2 is small, Then y^C
Rule F_2: If x_1 is medium and x_2 is medium, Then y^C
Rule F_3: If x_1 is high and x_2 is small, Then y^C
Rule F_4: If x_1 is high and x_2 is medium, Then y^C

3.2. Local rule selection

3.2.1. Functional generalisation

Through the use of a similarity measure, the heuristic method generates a set of interpretable fuzzy rules with respect to each existing crisp rule. However, the employment of all such preliminarily mapped fuzzy rules does not necessarily optimally mimic the capability of the original crisp rule. Unlike crisp rule-based environment, where an instance is only covered by one crisp rule, each instance may now match with multiple fuzzy rules to various degrees. Unfortunately, certain mapped fuzzy rules may be conflicting with each other, whilst certain rules may be rather similar with one another (resulting in duplications). These issues must be addressed, not just to increase computational efficiency but also to decrease potential model inconsistency and complexity.

A local rule selection procedure is proposed here to tackle these issues, by introducing the constraint of *functional generalisation*. This constraint imposes that in searching for a subset of initially mapped fuzzy rules to replace the full set of the (possibly inconsistent and/or redundant) preliminary rules, the rule subset must collectively generalise the capability of the original crisp rule from which they are mapped while avoiding or minimising inconsistency and redundancy.

Suppose that there are N crisp rules $C_j, j = 1, 2, \ldots, N,$ and that K_j preliminary fuzzy rules $F_{ji}, i = 1, 2, \ldots, K_j$ are mapped from C_j using the heuristic method. For each input pattern $\bar{x}^p \in E_{trn}$, the rule firing degree $\mu_{F_{ji}}(\bar{x}^p)$ with respect to the entire set of fuzzy rules F_{ji} is intuitively defined as the largest matching degree

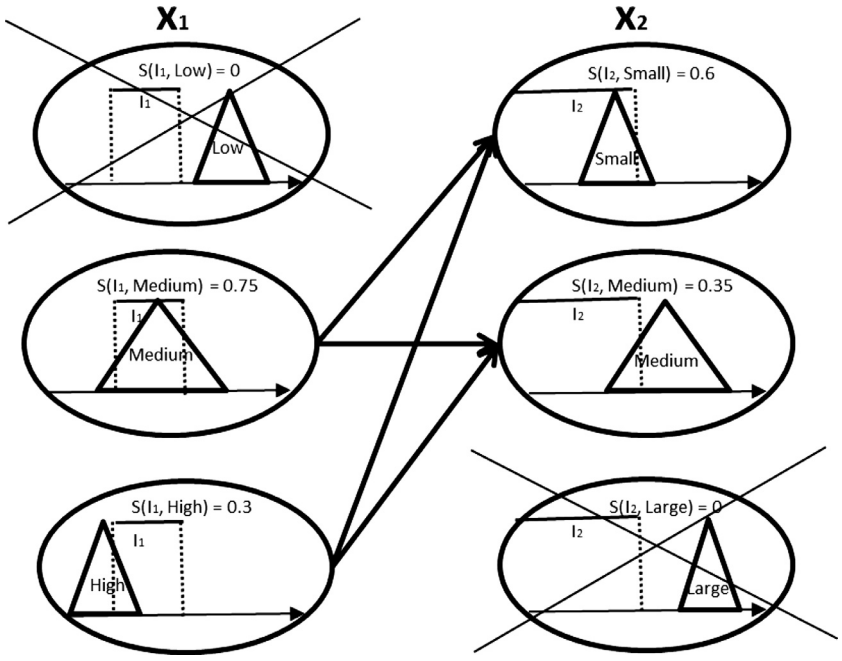

Fig. 1. Example on heuristic mapping.

amongst all:

$$\mu_{F_{ji}}(\bar{x}^p) = \max\{\mu_{F_{j1}}(\bar{x}^p), \ldots, \mu_{F_{ji}}(\bar{x}^p), \ldots, \mu_{F_{jK_j}}(\bar{x}^p)\} \tag{9}$$

Let E_j denote the set of instances selected to measure the quality of a selected subset of fuzzy rules $F_{ji'}, i' = 1, 2, \ldots, S_j, S_j \leq K_j$, which satisfies the following:

$$E_j = \{\bar{x}^p | \mu_{F_{ji}}(\bar{x}^p) > 0, \bar{x}^p \in E_{trn}, i = 1, \ldots, K_j\} \tag{10}$$

To ensure the desired functional generalisation, there are five cases to consider regarding the different instances of a given E_j:

(1) Instances that are covered and correctly classified by the original crisp rule C_j:

$$E_{j1} = \{\bar{x}^p | y^p = y^{C_j}, \mu_{C_j}(\bar{x}^p) = 1, \bar{x}^p \in E_j\} \tag{11}$$

where y^p is the underlying label of the instance $\bar{x}^p$, and y^{C_j} is the rule consequent of C_j. It is desirable to maximise the firing degrees over these instances when using the selected fuzzy rules, by imposing the constraint that such instances be still correctly classified, while avoiding influence from other mapped fuzzy rules, especially those whose rule consequents are inconsistent with the selected rules.

(2) Instances that are covered, but wrongly classified by C_j:

$$E_{j2} = \{\bar{x}^p | y^p \neq y^{C_j}, \mu_{C_j}(\bar{x}^p) = 1, \bar{x}^p \in E_j\} \tag{12}$$

It is desirable to minimise the firing degrees over these instances when using the selected fuzzy rules, as much as possible, while improving the opportunity for them to be classified by other mapped fuzzy rules with consistent class labels.

(3) Instances that are not covered by the original crisp rule C_j, but by an alternative rule $C_{j'}$ with correct classification which happens to be of the same consequent as C_j, and that are matched to a certain extent with the fuzzy rules F_{ji} that are mapped from C_j with consistent classification:

$$E_{j3} = \{\bar{x}^p | y^p = y^{C_j} = y^{C_{j'}}, \mu_{C_{j'}}(\bar{x}^p) = 1, \quad j' \neq j, \bar{x}^p \in E_j\} \tag{13}$$

It is natural not to do anything in this case since the fuzzy rules mapped from C_j will provide the same correct class label as that inferred by certain other fuzzy rules mapped from the other original crisp rule $C_{j'}$.

(4) Instances that are otherwise regarded as the same as those in Case (iii), except that they are incorrectly classified by $C_{j'}$:

$$E_{j4} = \{\bar{x}^p | y^p = y^{C_j} \neq y^{C_{j'}}, \mu_{C_{j'}}(\bar{x}^p) = 1, \quad j' \neq j, \bar{x}^p \in E_j\} \tag{14}$$

In this case, it is desirable to maximise the firing degrees over these instances when using the fuzzy rules selected from those mapped from C_j, as much as possible, while providing additional support for those instances of Case (ii).

(5) Instances whose class labels are inconsistent with those of the original crisp rule C_j, but either they are correctly classified by an alternative rule $C_{j'}$ with a consistent rule consequent:

$$E_{j5a} = \{\bar{x}^p | y^{C_j} \neq y^p = y^{C_{j'}}, \mu_{C_{j'}}(\bar{x}^p) = 1, \quad j' \neq j, \bar{x}^p \in E_j\} \tag{15}$$

or they are incorrectly classified by an alternative rule $C_{j'}$:

$$E_{j5b} = \{\bar{x}^p | y^p \neq y^{C_j}, y^p \neq y^{C_{j'}}, \mu_{C_{j'}}(\bar{x}^p) = 1, \quad j' \neq j, \bar{x}^p \in E_j\} \tag{16}$$

It is desirable to minimise the firing degrees over these instances when using the selected fuzzy rules, as much as possible, given that the consequents of such fuzzy rules are not to be consistent with the true classes of these instances, while improving the opportunity for them to be matched with rules that are mapped from other crisp rules with correct classification. For simplicity in description later, introduce the notion of E_{j5} such that $E_{j5} = E_{j5a} \cup E_{j5b}$.

3.2.2. Search for subset of quality mapped rules

Given the above discussion, the quality $Q(F_{ji'})$ of a subset of the fuzzy rules $F_{ji'}, i' = 1, 2, \ldots, S_j, S_j \leq K_j$, selected from the K_j rules $F_{ji}, i = 1, 2, \ldots, K_j$, mapped from the preliminary crisp rule C_j in relation to the data set E_j, can be evaluated as follows:

$$Q(F_{ji'}) = \sum_i Q_{E_{ji}}(F_{ji'}) \tag{17}$$

where each of the $Q_{E_{ji}}(F_{ji'}) \in [0, 1], i = 1, 2, 4, 5,$ denotes the quality measure of the fuzzy rule subset computed over the data instances that belong to Case *i*. Note that Case 3 is not included due to its nature as indicated previously.

The component quality measures $Q_{E_{ji}}$ are defined using the following biased mean squared error, following that which is popularly adopted in conventional classification techniques:

$$Q_{E_{ji}}(F_{ji'}) = 1 - \frac{1}{|E_{ji}|} \sum_{\bar{x}^p \in E_{ji}} (\mu_{F_{ji'}}(\bar{x}^p) - \theta)^2, \tag{18}$$

where $|E_{ji}|$ is the cardinality of instances from Case i; $\mu_{F_{ji'}}(\bar{x}^p)$ denotes the largest matching degree of the instance $\bar{x}^p$ with the selected subset of fuzzy rules $F_{ji'}$; $\theta \in \{0.0, 1.0\}$ represents the desired value (depending on whether it is for maximisation or minimisation) regarding the instance $\bar{x}^p$, that is, $\theta = 1.0$ if $\bar{x}^p \in E_{j1} \cup E_{j4}$, $\theta = 0.0$ if $\bar{x}^p \in E_{j2} \cup E_{j5}$.

Following the above approach the generalisation capability of the selected fuzzy rules that are mapped from a given crisp rule C_j is assessed with regard to an equal weight over the five types of training data instance. This may not be the ideal in general because not only the number of instances from different types can vary, the matching degrees of individual instances are not the same either, where higher matching degrees ought to be considered contributing more to the overall quality than the lower ones.

To better address this issue, a weighted approach is taken here. In particular, the weight w_{ji} that is associated with an individual quality measure is specified as the ratio between the sum of the matching degrees of the instances belonging to that given type E_{ji} and the total of the matching degrees of all instances in E_j such that

$$w_{ji} = \frac{\sum_{\bar{x}^p \in E_{ji}} \mu_{F_{ji}}(\bar{x}^p)}{\sum_{i=1,2,4,5} \sum_{\bar{x}^p \in E_{ji}} \mu_{F_{ji}}(\bar{x}^p)} \tag{19}$$

where $\mu_{F_{ji}}(\bar{x}^p)$ is the matching degree of the instance $\bar{x}^p$ regarding all K_j preliminary fuzzy rules as defined in Eq. (9). In addition, to minimise the generation of potentially redundant rules, the following relative size $S(F_{ji'})$ of the resultant fuzzy rules is also factored into the overall quality measure:

$$S(F_{ji'}) = 1 - \frac{|F_{ji'}|}{|F_{ji}|} \tag{20}$$

Thus, the quality $Q(F_{ji'})$ of a selected subset of the fuzzy rules $F_{ji'}$ mapped from a given crisp rule C_j will be assessed as follows:

$$Q(F_{ji'}) = \sum_{i=1,2,4,5} w_{ji} Q_{E_{ji}}(F_{ji'}) + w_s S(F_{ji'}) \tag{21}$$

where $w_s \in [0, 1]$ is a parameter that allows for the adjustment of the relative contribution of the size of the subset of selected fuzzy rules towards the quality of that subset (which may be set to 1 by default for simplicity in implementation).

3.3. Tuning of interpretable fuzzy rule base

The above work ensures that a subset of fuzzy rules can be selected that collectively generalise a given crisp rule. However, globally, the combination of all such locally selected fuzzy rules does not necessarily result in an optimal and compact interpretable rule base, especially from the ruleset complexity viewpoint. Although each subset of rules may be optimised separately, the quality of any neighbouring subsets which share antecedent variables may be deteriorated if they are not optimised at the same time (e.g., due to the possible generation of conflicting rule sets). The overall performance of the entire rule base is thus unpredictable when all crisp rules are mapped simultaneously. With the aim to obtain a compact ruleset with high performance, when given all of the selected fuzzy rules in response to all existing crisp rules, a technique is therefore required to search for an optimal set of fuzzy rules globally.

Genetic algorithms (GAs) are employed in this work to implement the required global search owing to their practical popularity and conceptual simplicity, instead of more powerful multi-objective GAs (MOGAs) [30]. Of course, MOGAs and other stochastic population-based techniques may be adopted as alternative, if preferred, which should help strengthen the performance of this work. In implementation within this work, the GA used adopts Pittsburgh style encapsulation, whereby the combination of all selected fuzzy rules returned by the local rule selection process are encoded within a single chromosome, where individuals of the first population are initialised with an exact copy of the selected fuzzy rules.

Generally speaking, in applying GAs, a set of possible solutions are represented as chromosomes, with better emerging solutions more likely to be selected as offsprings according to their fitness, where new solutions are generated mainly based on crossover and mutation operators. In order to allow more flexibility for ruleset tuning, each encoded fuzzy rule is assumed to always include n antecedents, with a *don't care* label in place of void in the corresponding variable location within the rule. Obviously, an emerging rule will be eliminated if *don't care* appears as the value for all antecedent variables. In so doing, for a problem involving an n-dimensional pattern space, each variable $x_i, i \in \{1, 2, \ldots, n\}$ may take any fuzzy set from its domain $\{D_0, D_1, \ldots, D_{d_i}\}$ (whose cardinality is d_i), with D_0 representing the notion of *don't care* (that has a specifically fixed membership value of 1).

Recall that the ultimate goal of this tuning process is to obtain an accurate fuzzy rule base that is interpretable in terms of both semantics and complexity. As the semantic interpretability is already ensured by the consistent use of predefined fuzzy sets, the fitness function takes both the accuracy and complexity of a resultant fuzzy rule base into account, such that

$$Q = Q_p - w_i Q_i \tag{22}$$

where Q_p measures the performance of the resultant rule base, defined as the accuracy rate of correctly classified instances; Q_i measures the structural complexity of the rule base, defined as the size of the resulting rule base, penalising rule base with a large number of rules or rules of many compound conditions; and w_i is a weighting factor to balance the expected contributions of the two quality indicators. As such, this work follows a conceptually simple method that converts multiple objectives into a compound single objective.

3.4. Summary and complexity analysis

Given a set of crisp rules $\{C_j | j = 1, 2, \ldots, N\}$ (provided by domain experts or returned by a certain existing data-driven crisp rule learner), and a fixed linguistic term set with underlying semantics defined as fuzzy sets reflecting the domain expertise, the process of generating an interpretable fuzzy rule base can be summarised into the following three-stage process, as outlined in Fig. 2.

Stage 1 Mapping crisp rules into interpretable fuzzy rules. For each crisp rule C_j:

(a) Generate the (sub-)data set E_j^i relevant to each antecedent variable x_i.
(b) Calculate similarity between the crisp interval I_{ji} and each of the predefined fuzzy set D_{ji} of x_i.
(c) Retain those fuzzy sets whose similarity values surpass user-defined threshold η.

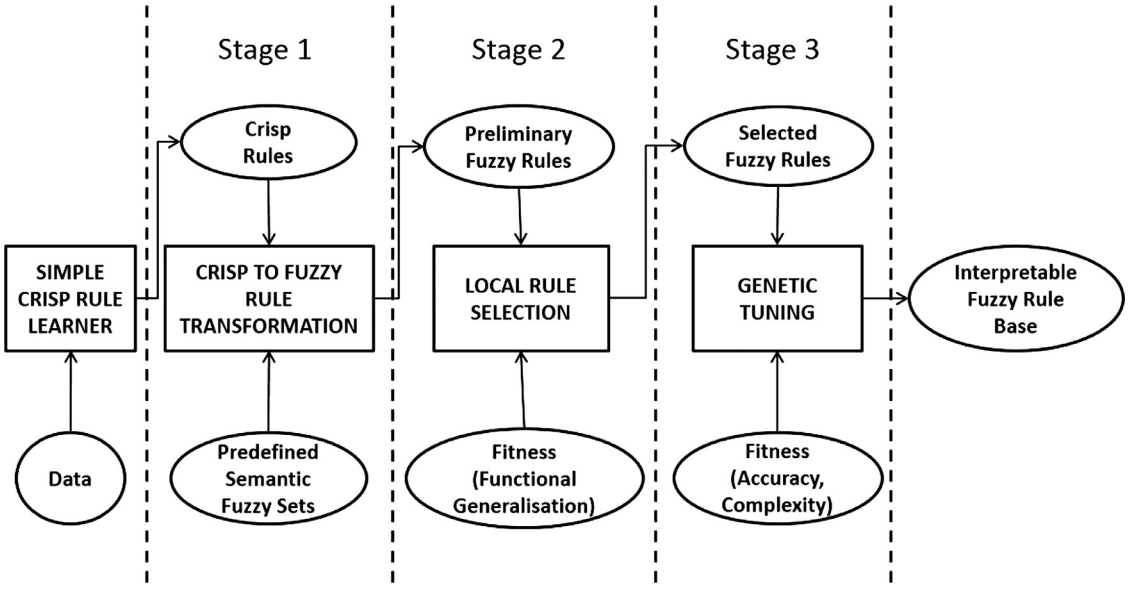

Fig. 2. Generation of accurate and interpretable fuzzy model from a crisp rule learner: Three stages

(d) Create a set of emerging interpretable fuzzy rules $F_{ji}, i = 1, 2, \ldots, K_j$ using the heuristic method.

The cost incurred in this stage to generate the initial sets of fuzzy rules is $O(N \times N_{intl} \times d)$, where N denotes the number of given crisp rules, N_{intl} is the maximum number of the existing crisp intervals for any crisp rule, and d is the maximum number of predefined fuzzy sets for any attribute. In practice, N_{intl} is set to a small number to allow for more general rules [29] whilst d is not large, which is typically at most 9 owing to psychological theory for the learned rules to be interpretable (although in the experimentation later, this may be set to 14 in an effort to demonstrate that the proposed method works even with variable domains more complex than usual).

Stage 2 Selecting mapped fuzzy rules with functional generalisation. For each set of fuzzy rules $F_{ji}, i = 1, 2, \ldots, K_j$ mapped from C_j:

(a) Categorise instances from E_j into five types.
(b) Compute weights for each type.
(c) Obtain a locally optimal selected subset of fuzzy rules $F_{ji'}, i' = 1, 2, \ldots, S_j, S_j \leq K_j$ with functional generalisation (which is also implemented with a simple GA in this work).

In terms of computational effort to implement this stage, the cardinality of possible fuzzy rules generated in response to each crisp rule is bounded by $N_{intl} \times T$, where T is the maximum number of similar fuzzy sets that are allowed per crisp interval. In practice, as with N_{intl}, T is set to a small number to avoid potentially generating too many redundant rules. For each crisp rule, the cost for rule evaluation over a subset of initially mapped fuzzy rules is bounded by $2^{N_{intl} \times T}$. The total computational effort at this stage is therefore, $O(N \times 2^{N_{intl} \times T})$, which can be practically resolved by GA given that N_{intl} and T are both a small number.

Stage 3 Computing a compact and accurate fuzzy rule base with a GA.

(a) Encode all locally optimised fuzzy rules together in Pittsburgh style.
(b) Optimise the interpretable fuzzy rule base, with performance and complexity jointly encoded as fitness function.

Suppose that the cardinality of the family of all selected fuzzy rules is N_r, then, the cost in implementing this stage for the final generic tuning is $O(d^n \times N_r)$, where n is the number of antecedent attributes in the domain. In practice, as the outcome of Stage 2 has already provided a good solution and d is not large, the GA often converges very quickly here (which is also supported by experimental results as to be shown in Section 4.4).

Finally, note that at the end of each stage, appropriate conventional rule-pruning mechanisms may be employed if desired, but this is beyond the scope of this paper.

4. Experimentation on benchmark datasets

Systematic experiments using benchmark data sets are reported here to demonstrate the efficacy of the proposed approach. Section 4.1 introduces the experimental setup. Section 4.2 shows the generation of interpretable fuzzy rules, which are initialised from crisp rules generated by two distinct learning mechanisms, and compares the generated rules with those directly fuzzified by the use of the popular FURIA algorithm [29]. Section 4.3 compares performance of the generated rule bases with alternative fuzzy rule-based learning classifiers that only use fixed and predefined fuzzy sets, with rule bases complexity analyses as shown in Section 4.4. For completeness, Section 4.5 compares the proposed work with non-fuzzy-rule-based learning approaches. Section 4.6 investigates the effect of local rule selection in relation to functional generalisation.

4.1. Experimental setup

To demonstrate the proposed approach at work, experiments are performed on 16 real-valued UCI benchmark data sets [31]. A summary of the characteristics of these data sets is given in Table 1. Stratified tenfold cross-validation (10-CV) is employed for result validation. In 10-CV, a given data set is partitioned into ten subsets. Of the ten, nine subsets are used to perform training, where the proposed approach is used to generate an interpretable fuzzy rule base, and the remaining single subset is retained as the testing data for assessing the learned classifier's performance. This cross-validation process is then repeated ten times in order to lessen the impact of random factors; results of these 10×10 cross-validations are then averaged to produce each final

Table 1
Summary of data sets used.

Data set	Attribute no.	Instance no.	Class no.
appendicitis	7	106	2
banknote	4	1372	2
blood	4	748	2
breast-cancer	9	699	2
column-2C	6	310	2
column-3C	6	310	3
ionosphere	33	230	2
iris	4	150	3
liver-disorders	6	345	2
mammographic	5	961	2
new-thyroid	5	215	3
parkinsons	22	195	2
pima-diabetes	8	768	2
seeds	7	210	3
sonar	60	208	2
wdbc	30	569	2

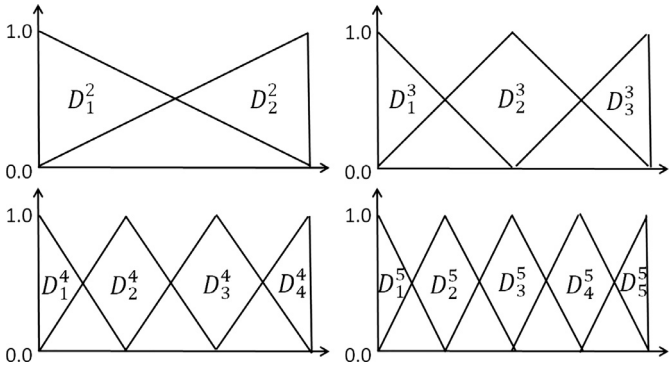

Fig. 3. Partitioning of pattern space.

experimental outcome reported below (except for the particular investigation into the effect of local rule selection as reported in Section 4.6).

For fair and systematic comparison, fixed and uniformly divided fuzzy sets are used in the experiments across all datasets. As the partition granularity for each variable is unknown in advance, in this work, without any bias and for simplicity, four sub-types of homogeneous fuzzy partition with uniformly divided triangular MFs are employed, as shown in Fig. 3, where D_j^i denotes the j-th fuzzy set generated from the uniform division of the input domain into i partitions. That is, each antecedent variable may take one fuzzy set from the domain:

$$\{D_1^2, D_2^2 D_1^3, D_2^3, D_3^3, D_1^4, D_2^4, D_3^4, D_4^4, D_1^5, D_2^5, D_3^5, D_4^5, D_5^5\}$$

(in addition to the value that stands for *don't care*). Given such underlying value domains, 4 bits are required for encoding each variable in the binary encoded chromosomes, with 0000 and 1111 reserved for the *don't care* label, and the rest for the 14 distinct fuzzy sets. The total length of a chromosome required is $4nN_r$, where N_r is the cardinality of the family of all selected fuzzy rules after Stage 2 of the learning process. The fitness function is defined as given in Eq. (22). Each implemented GA utilises the strategy of steady-state with elitism selection.

As the main aim of this investigation is to examine the efficacy of the proposed approach for the acquisition of an interpretable rule base for the development of fuzzy KBSs, instead of the performance of a GA itself, only the basic version of GA is used in the experiments. Each implemented GA utilises the rank-based selection with the steady-state and elitism strategy, where the simple one-point mutation is adopted as mutation operator. The parameter specification for GA is not purposefully adjusted and therefore, the experimental results could be further improved where a more sophisticated version of GA is employed with carefully modified parameters. In particular, GAs with the same parameter specification as detailed in Table 2 are applied to generate fuzzy rules that are initialised by two distinct crisp rule-based learning mechanisms to facilitate comparison. Specification of other parameters involved in the proposed approach and alternative learning classifiers is summarised in Table 3. Note that the implementation of the compared approaches can be found in WEKA [32] or KEEL [33].

4.2. Generating fuzzy rules with C4.5 and unordered RIPPER

Two highly popular crisp rule-based classifier learners are each employed here to act as the initial crisp rule generator to enrich the comparison. These are C4.5, a classical decision tree learning algorithm, and an unordered version of RIPPER (UR) [29]. Comparison is also made with FURIA [29], commonly served as the benchmark that greedily transforms crisp rules into fuzzy rules by fitting initially generated crisp intervals into parameterised trapezoid MFs, where C4.5 and UR are also separately used as the initial rule generator. For conciseness, the resulting learned rule sets are shorthanded as C45-IFRC and UR-IFRC, for C4.5- and UR-initialised interpretable fuzzy rule-based classifiers, and as C45-FURIA and UR-FURIA for C4.5- and UR-initialised FURIA, respectively. Note that UR-FURIA is the exact FURIA algorithm itself that converts UR rules directly into fuzzy rules, and is herein renamed purely for meeting the eyes.

Table 4 presents the results with C4.5 used as the initial rule generator, where the top performer in terms of classification accuracy is highlighted in boldface for each data set, and pair-wise t-test ($p = 0.05$) results are identified to reflect their statistical significance. As can be seen, performance improvement using the present approach is statistically very significant with 10 wins, 4 ties and only 2 losses. In particular, C45-IFRC works well generally across the data sets with a different dimensionality, achieving 12 top results out of 16. Superiority in performance of the fuzzy rules produced using the proposed approach over those generated by FURIA is also statistically reflected in the last column of Table. 4, where C45-IFRC clearly beats C45-FURIA with 7 wins, 7 ties and only 2 losses. In contrast, the performance of the fuzzy rule bases generated by FURIA is even worse than its original crisp counterpart, with t-test results barely being equal.

Table 5 lists the results with unordered RIPPER used as the initial rule generator. The performance improvement owing to the use of the proposed algorithm is also significant with 8 wins, 6 ties and 2 losses, albeit having 2 wins fewer than the number achieved by FURIA. Different behaviours of FURIA in fuzzifying two different types of crisp rule bases (returned by C4.5 and UR, respectively) can be observed. This is because UR works by searching for fuzzified outcomes for one antecedent variable at a time in a brute-force way, thereby meeting the underlying strategy taken by FURIA, whilst C4.5 works over all individual attributes by one go. Never-

Table 2
Parameter specification of GA.

Stage 2	$w_s = 0.1, Pop = 100, P_c = 0.95, P_m = 0.005, maxItr = 100, itr\_no\_improve = 10$
Stage 3	$w_i = 0, Pop = 100, P_c = 0.95, P_m = 0.005, maxItr = 500, itr\_no\_improve = 30$

Table 3
Parameter specification of learning classifiers.

Approach	Parameter Specification
PTTD	$\epsilon = 0.0025, numCands = 5, maxDepth = 0$
GP-COACH	$Labels = 5, Eval = 20000, Pop = 200, \alpha = 0.7, P_c = 0.5, P_m = 0.2,\ P_{dp} = 0.15, P_i = 0.15, Tournament = 2, w_1 = 0.8, w_2 = w_3 = 0.05, w_4 = 0.1$
SLAVE2	$Pop = 20, Iter_{\text{change}} = 500, P_{bm} = 0.5, P_{bc} = 0.1, P_{rm} = 1.0, P_{rc} = 0.2, \lambda = 0.8$
MOGUL	$Labels = 5, \omega = 0.05, K = 0.1, \epsilon = 1.5, repeat\_rules = 1, rule\_type = 2,\ Iter_{\text{selection}} = 500, Pop_{\text{selection}} = 61, \tau = 1.5, \beta = 0.5, P_{cs} = 0.6, P_{ms} = 0.1,$ $Iter_{\text{tuning}} = 1000, Pop_{\text{tuning}} = 61, a = 0.35, b = 5, P_{ct} = 0.6, P_{mt} = 0.1$
FH-GBML	$Rules = 30, Sets = 200, Gens = 1000, P_c = 0.9, P_{\text{dont-care}} = 0.5, P_{\text{michigan}} = 0.5$
SGERD	$Q_{\text{rules}} = 0$(calculate heuristically), $RuleEval = 2$
QSBA	$Labels = 5, thres = 0.7, Tnorm =$ Algebraic
C4.5	$Pruned =$ yes, $confidence = 0.25, minNumObj = 2, numFolds = 3, reduced\_error\_pruned =$ yes
RIPPER	$Pruning = yes, Folds = 3, N_{\text{optimisations}} = 2$
NB	default
SMO	$c = 1.0, \epsilon = 1.0 \times 10^{-12}, tolerance = 0.001$
IBk	$kNN = 1, search\_algorithm =$ linear search, $window = 0$
FRNN	$kNN = 10, TNorm =$ KD, $Implicator =$ KD, $Similarity = 1$
NFC	$epoch = 100, \sigma = 5.0e^{-5}, \lambda = 5.0e^{-7}$
C45-IFRC	$maxDepth = 3, T = 3, \eta = 0, w_i = 0$
UR-IFRC	$maxDepth = 5, T = 3, \eta = 0, w_i = 0$

Table 4
Comparison of classification accuracy (%) with C4.5 as initial rule generator, where v, -, and * indicate statistically better, same, and worse against C4.5.

Data set	C4.5	1(C45-IFRC)	2(C45-FURIA)	1 vs. 2
appendicitis	82.79 ± 1.78	**84.34 ± 2.63** (v)	73.47 ± 13.93 (*)	(v)
banknote	97.96 ± 0.45	**98.63 ± 0.34** (v)	98.16 ± 0.37 (v)	(v)
blood	76.89 ± 0.82	**77.53 ± 0.48** (v)	59.59 ± 12.28 (*)	(v)
breast-cancer	94.13 ± 0.65	**95.15 ± 0.68** (v)	94.81 ± 0.55 (v)	(-)
column-2C	79.52 ± 1.74	**80.16 ± 2.16** (-)	79.7 ± 1.77 (-)	(-)
column-3C	79.81 ± 2.05	77.51 ± 1.9 (*)	**79.93 ± 2.17** (-)	(*)
ionosphere	**87.00 ± 1.19**	86.8 ± 0.72 (-)	86.62 ± 1.26 (*)	(-)
iris	93.33 ± 1.3	**95.32 ± 0.54** (v)	93.33 ± 1.17 (-)	(v)
liver-disorders	63.08 ± 2.45	**64.83 ± 1.64** (v)	63.16 ± 2.13 (-)	(v)
mammographic	**82.03 ± 0.66**	79.13 ± 0.79 (*)	81.49 ± 1.04 (-)	(*)
new-thyroid	91.35 ± 1.52	**91.88 ± 1.20** (-)	91.54 ± 1.39 (-)	(-)
parkinsons	**84.48 ± 2.26**	84.33 ± 1.11 (-)	84.42 ± 2.24 (-)	(-)
pima-diabetes	73.89 ± 0.77	**75.05 ± 0.89** (v)	74.22 ± 0.81 (v)	(v)
seeds	90.38 ± 1.1	**91.37 ± 1.25** (v)	90.61 ± 0.95 (-)	(-)
sonar	70.23 ± 3.36	**72.59 ± 4.21** (v)	70.67 ± 3.44 (-)	(v)
wdbc	93.76 ± 0.64	**94.30 ± 0.53** (v)	93.87 ± 0.64 (-)	(-)
Summary (*/-/v)	83.789	**84.308** (2/4/10)	82.224 (3/10/3)	(2/7/7)

Table 5
Classification accuracy (%) with UR as initial rule generator, where v, -, and * indicate statistically better, same, and worse against UR.

Data set	UR	1(UR-IFRC)	2(UR-FURIA)	1 vs. 2
appendicitis	85.79 ± 1.8	**86.83 ± 1.70** (v)	85.8 ± 1.92 (-)	(v)
banknote	98.4 ± 0.22	98.75 ± 0.22 (v)	**99.12 ± 0.22** (v)	(*)
blood	**78.02 ± 0.55**	77.82 ± 1.11 (-)	**78.02 ± 0.55** (-)	(-)
breast-cancer	94.16 ± 0.47	**95.90 ± 0.39** (v)	94.96 ± 0.41 (v)	(v)
column-2C	81.9 ± 1.9	81 ± 2.07 (-)	**82.39 ± 1.67** (v)	(*)
column-3C	75.54 ± 0.88	**78.76 ± 1.68** (v)	77.52 ± 1.57 (v)	(v)
ionosphere	86.76 ± 1.07	85.58 ± 1.84 (-)	**87.35 ± 1.37** (-)	(*)
iris	92.59 ± 1.23	**95.59 ± 0.56** (v)	94.33 ± 0.72 (v)	(v)
liver-disorders	66.97 ± 2.18	64.86 ± 2.09 (*)	**68.79 ± 2.00** (v)	(*)
mammographic	82.31 ± 0.34	78.46 ± 1.09 (*)	**82.53 ± 0.49** (-)	(*)
new-thyroid	94.28 ± 0.72	94.29 ± 0.85 (-)	**94.84 ± 0.86** (v)	(-)
parkinsons	88.3 ± 1.98	87.02 ± 1.34 (-)	**89.87 ± 1.32** (v)	(*)
pima-diabetes	74.82 ± 0.87	**75.27 ± 0.69** (-)	74.93 ± 1.03 (-)	(-)
seeds	90.48 ± 0.78	**92.66 ± 1.52** (v)	92.05 ± 0.68 (v)	(-)
sonar	74.82 ± 2.26	**77.39 ± 1.98** (v)	75.49 ± 2.09 (-)	(v)
wdbc	94.44 ± 0.55	94.98 ± 0.73 (v)	**94.99 ± 0.45** (v)	(-)
Summary (*/-/v)	84.974	85.323 (2/6/8)	**85.811** (0/6/10)	(6/5/5)

theless, the proposed approach is shown to be able to work with both strategies, leading to significant performance improvements.

As each of the original crisp rules points to different places where potentially desirable fuzzy rules may exist, the quality of preliminary crisp rules has an obvious impact upon the final generated fuzzy rules, as illustrated above. Thus, any direct attempt to compare the performances between the two fuzzy rule bases produced by C45-IFRC and UR-IFRC makes little sense, given their very different head start points. What is important is that they both achieve improved performances using only predefined fuzzy sets, producing models of inherent interpretability.

Table 6
Comparison on classification accuracy (%) against interpretable fuzzy classifiers.

Data Set	C45-IFRC	UR-IFRC	PTTD	GP-COACH	SLAVE2	MOGUL	FH-GBML	SGERD	QSBA
appendicitis	84.34 ± 2.63	**86.83 ± 1.70**	86.66 ± 0.89	86.27 ± 1.81	84.36 ± 1.52	76.97 ± 2.47	84.33 ± 2.06	85.04 ± 1.01	86.48 ± 0.95
banknote	98.63 ± 0.34	98.75 ± 0.22	84.52 ± 0.15	91.59 ± 0.65	91.63 ± 0.17	**98.99 ± 0.22**	98.36 ± 0.41	84.20 ± 0.36	82.19 ± 0.31
blood	77.53 ± 0.48	77.82 ± 1.11	77.42 ± 0.13	76.17 ± 0.24	76.51 ± 0.14	**78.18 ± 0.56**	77.04 ± 0.30	76.22 ± 0.18	66.58 ± 1.32
breast-cancer	95.15 ± 0.68	95.90 ± 0.39	95.35 ± 0.23	95.78 ± 0.45	96.15 ± 0.35	75.46 ± 0.81	**96.21 ± 0.56**	93.49 ± 0.36	95.65 ± 0.16
column-2c	80.16 ± 2.16	81.00 ± 2.07	74.65 ± 1.48	75.52 ± 1.12	79.00 ± 1.04	77.03 ± 1.34	**81.35 ± 1.84**	70.00 ± 1.22	69.42 ± 0.34
column-3c	77.51 ± 1.90	**78.76 ± 1.68**	75.16 ± 0.79	74.65 ± 2.16	74.94 ± 0.71	75.45 ± 1.78	77.87 ± 1.31	70.77 ± 1.25	70.94 ± 0.42
ionosphere	86.80 ± 0.72	85.58 ± 1.84	74.04 ± 1.16	**90.09 ± 0.96**	89.61 ± 1.35	31.13 ± 1.11	48.13 ± 2.35	73.65 ± 1.77	82.96 ± 0.94
iris	95.32 ± 0.54	95.59 ± 0.56	95.60 ± 0.34	**97.67 ± 0.35**	96.60 ± 0.38	93.33 ± 1.40	94.20 ± 1.18	94.27 ± 1.00	91.67 ± 0.35
liver-disorders	64.83 ± 1.64	64.86 ± 2.09	**67.75 ± 1.52**	58.99 ± 0.71	60.76 ± 0.71	58.77 ± 2.75	65.89 ± 1.77	59.01 ± 1.10	57.69 ± 1.02
mammographic	79.13 ± 0.79	78.46 ± 1.09	76.23 ± 0.44	78.95 ± 0.42	78.58 ± 0.62	78.85 ± 0.73	**80.87 ± 0.64**	77.39 ± 0.20	80.58 ± 0.26
new-thyroid	91.88 ± 1.20	**94.29 ± 0.85**	88.75 ± 0.53	91.78 ± 0.65	91.56 ± 0.50	93.50 ± 1.18	92.63 ± 0.91	87.23 ± 0.58	93.12 ± 0.61
parkinsons	84.33 ± 1.11	87.02 ± 1.34	85.03 ± 0.71	**87.27 ± 1.03**	86.82 ± 1.25	62.38 ± 2.71	81.26 ± 1.10	82.28 ± 1.53	81.07 ± 1.22
pima-diabetes	75.05 ± 0.89	75.27 ± 0.69	74.13 ± 0.36	75.13 ± 0.87	**75.38 ± 0.71**	71.26 ± 0.77	73.72 ± 1.03	70.17 ± 0.69	73.45 ± 0.65
seeds	91.37 ± 1.25	**92.66 ± 1.52**	89.43 ± 1.00	91.67 ± 1.19	90.00 ± 1.31	91.65 ± 1.23	90.76 ± 1.71	86.52 ± 0.87	81.57 ± 0.64
sonar	72.59 ± 4.21	77.39 ± 1.98	67.77 ± 1.57	**78.86 ± 3.01**	78.07 ± 1.83	5.91 ± 1.34	45.06 ± 3.17	70.09 ± 2.38	74.44 ± 0.82
wdbc	94.30 ± 0.53	**94.98 ± 0.73**	93.25 ± 0.54	94.41 ± 0.50	94.66 ± 0.44	81.62 ± 0.92	90.47 ± 0.96	91.86 ± 0.67	91.35 ± 0.30
Summary	84.308	**85.323**	81.609	84.049	84.039	71.905	79.885	79.512	79.946

Table 7
Comparison against fuzzy rule-based classifiers, where v, -, and * indicate statistically better, same, and worse classification performance against the proposed approach.

Data Sets	C45-IFRC							UR-IFRC						
	PTTD	GP-COACH	SLAVE2	MOGUL	FH-GBML	SGERD	QSBA	PTTD	GP-COACH	SLAVE2	MOGUL	FH-GBML	SGERD	QSBA
appendicitis	v	-	-	*	-	-	v	-	-	*	*	*	*	-
banknote-authentication	*	*	*	v	*	*	*	*	*	*	v	*	*	*
blood	-	*	*	v	*	*	*	-	*	*	-	*	*	*
breast-cancer-wisconsin	-	v	-	*	v	*	v	*	-	v	*	-	*	-
column-2C	*	*	*	*	-	*	*	*	*	*	*	-	*	*
column-3C	*	*	v	*	-	*	*	*	*	*	*	-	*	*
ionosphere	*	v	v	*	*	*	*	*	v	v	*	*	*	*
iris	-	v	*	*	*	*	*	-	v	v	*	*	*	*
liver-disorders	v	*	-	*	-	*	*	v	*	*	*	-	*	*
mammographic	*	-	-	-	v	*	v	*	-	-	-	v	*	v
new-thyroid	*	-	v	v	-	*	v	*	*	*	*	*	*	*
parkinsons	-	v	-	*	*	*	*	*	-	-	*	*	*	*
pima-diabetes	*	-	*	*	*	*	*	*	-	-	*	*	*	*
seeds	*	-	-	-	-	*	*	*	-	*	*	*	*	*
sonar	*	v	v	*	*	*	-	*	-	-	*	*	*	*
wdbc	*	-	v	*	*	*	*	*	*	-	*	*	*	*
Summary (*/-/v)	(10/4/2)	(5/6/5)	(5/6/5)	(11/2/3)	(8/6/2)	(15/1/0)	(11/1/4)	(12/3/1)	(7/7/2)	(8/5/3)	(13/2/1)	(11/4/1)	(16/0/0)	(13/2/1)

4.3. Comparison with alternative interpretable fuzzy learning classifiers

Performance of both classifiers implemented using the two resultant fuzzy rule bases (i.e., C45-IFRC and UR-IFRC) is compared against 7 alternative fuzzy learning classifiers which also induce interpretable fuzzy rules with only fixed and uniformly divided quantity space. The results on classification accuracy are summarised in Table 6, and the corresponding *t*-test outcomes are shown in Table 7. The compared algorithms are as follows:

PTTD [7,25] is a fuzzy pattern-tree learning classifier, which is composed of an ensemble of pattern trees, one for each class. A pattern tree is a hierarchical, tree-like structure, whose inner nodes are marked with generalised logical operators and leaf nodes are associated with fuzzy predicates on the input attributes.

GP-COACH [26] is a genetic programming-based learning approach, which learns rules of a disjunctive normal form with a coding scheme that represents one rule per tree. GP-COACH uses a token competition mechanism to maintain the diversity of the population and this obliges the rules to compete and cooperate among themselves in order to obtain a compact set of fuzzy rules.

SLAVE2 [23,24] is an improved version of SLAVE, which learns rules of a disjunctive normal form through an iterative induction algorithm. SLAVE2 includes more information in the process of learning individuals rules, utilising the proposed calculus of the positive and negative examples, as well as new fitness functions and genetic operators.

FH-GBML [21,22] is a hybrid algorithm of two fuzzy genetics-based approaches for designing FRBCSs. It uses the Pittsburgh style to encode a set of fuzzy rules as an individual, while using the Michigan style to generate new rules to conduct heuristic mutation for partially modifying each rule set. As such, it exploits the advantages of both Michigan and Pittsburgh approaches.

SGERD [27] offers a novel steady-state GA-based algorithm to extract a compact set of fuzzy rules. The selection mechanism is non-random, such that only the best individuals can survive. It also makes use of rule and data dependent parameters, as well as an enhancing function to assess the candidate rules more effectively before selection.

MOGUL [34] is a method that learns genetic fuzzy rule-based systems through three stages. An initial rule set is first obtained via a genetic iterative process. This is followed by an additional genetic simplification procedure, which is then followed by a fine-tuning process for the emerging rules.

QSBA [15,35] is a fuzzy subsethood-based rule modelling mechanism. Continuous fuzzy quantifiers are used together with predefined membership functions in constructing learned rules. It works based on an iterative process of checking through the subsumption relationships amongst all concepts in the domain.

Table 8
Comparison on model complexity against fuzzy rule-based classifiers.

Data Sets	C45-IFRC		UR-IFRC		PTTD		GP-COACH		SLAVE2		MOGUL		FH-GBML		SGERD		QSBA	
	Rul	Cond	Rul	Cond	Rul	Cond	Rul	Cond	Rul	Cond	Rul	Cond	Rul	Cond	Rul	Cond	Rul	Cond
appendicitis	2.76	1.97	5.17	2.00	2.00	1.50	8.71	7.26	5.56	2.94	47.20	7.00	18.61	4.86	2.48	2.00	2.00	35.00
banknote-authentication	28.59	2.83	35.42	2.81	2.00	0.57	8.27	4.41	6.80	2.60	117.86	4.00	25.50	3.46	3.30	2.00	2.00	20.00
blood	7.51	2.67	7.35	2.25	2.00	1.50	7.38	3.92	4.19	1.95	129.11	4.00	17.55	3.41	2.69	2.00	2.00	20.00
breast-cancer-wisconsin	18.42	3.63	76.15	4.32	2.00	1.50	11.66	4.75	14.87	4.04	375.10	9.00	22.64	5.62	2.18	1.53	2.00	45.00
column-2C	8.12	3.36	14.8	3.14	2.00	1.50	4.70	4.99	5.65	2.96	157.75	6.00	18.88	4.74	2.98	2.00	2.00	30.00
column-3C	7.6	3.21	15.98	3.68	3.00	3.00	8.56	5.28	6.66	3.98	202.97	6.00	17.55	4.71	3.63	2.00	3.00	30.00
ionosphere	7.05	3.54	24.01	6.77	2.00	1.50	11.45	12.38	14.85	3.96	122.00	33.00	20.06	16.27	4.14	1.39	2.00	165.00
iris	4.35	1.45	8.73	2.12	3.00	3.00	3.04	2.14	5.86	1.72	68.70	4.00	23.18	3.24	3.63	1.99	3.00	20.00
liver-disorders	14.19	3.52	27.84	3.68	2.00	1.50	16.51	7.61	9.13	4.34	221.66	6.00	21.27	4.85	2.84	2.00	2.00	30.00
mammographic	10.9	3.15	5.46	2.79	2.00	1.50	13.73	3.78	8.58	2.95	639.79	5.00	23.75	3.75	2.02	2.00	2.00	25.00
new-thyroid	8.77	2.6	14.73	2.6	3.00	3.00	7.02	2.87	7.57	2.26	105.37	5.00	17.76	1.48	3.26	2.00	3.00	25.00
parkinsons	12.48	4.93	20.35	5.07	2.00	1.50	12.41	8.99	9.24	5.02	105.00	22.00	14.80	11.81	2.47	2.00	2.00	110.00
pima-diabetes	14.66	3.19	24.01	4.08	2.00	1.50	62.17	11.19	15.16	4.84	417.60	8.00	22.88	5.49	3.71	2.00	2.00	40.00
seeds	9.74	2.55	24.04	2.77	3.00	3.00	12.75	5.14	11.35	3.45	137.45	7.00	19.93	5.03	3.91	2.00	3.00	35.00
sonar	16.99	7.68	38.73	4.94	2.00	1.50	33.15	15.56	21.76	6.23	103.80	60.00	13.73	31.12	3.07	2.00	2.00	300.00
wdbc	13.4	4.98	43.83	8.55	2.00	1.50	9.16	4.74	9.16	5.60	281.90	30.00	16.53	15.22	3.57	2.00	2.00	150.00
Average	11.596	3.454	24.163	3.848	2.250	1.817	14.417	6.562	9.774	3.678	202.079	13.500	19.664	7.817	3.118	1.932	2.250	67.500

4.3.1. C45-IFRC vs. alternatives

Regarding individual data sets C45-IFRC does not always appear to be the top performer. However, its averaged performance across all tested datasets is higher than that achieved by any of the seven alternatives. In terms of statistical *t*-test, it ties with the two (GP-COACH and SLAVE2) and significantly beats the other five (e.g., C45-IFRC has 15 wins, 1 tie and no losses as compared to SGERD). Yet, GP-COACH and SLAVE2 learn fuzzy rules involving the use of disjunctive norm of fuzzy sets, i.e., they allow multiple fuzzy sets to be compounded to describe a single domain variable. This not only greatly expands the solution search space, but also causes the learned rules to become more complicated and hence less comprehensible.

4.3.2. UR-IFRC vs. alternatives

The performance of UR-IFRC is even more superior than C45-IFRC in terms of their relative performance against the seven alternatives. Again, it has achieved the best averaged accuracy amongst all. This is further supported with statistically significant better results throughout, even beating GP-COACH and SLAVE2, the two best performers amongst the seven, with substantially more wins than losses.

4.4. Model complexity

Table 8 presents an empirical analysis of the complexity of learned interpretable fuzzy rule bases, in terms of average number of antecedent conditions (*Cond*) per fuzzy rule, and average number of rules (*Rul*) per rule base.

For *Cond*, PTTD and SGERD return the most compact rules, both learning fuzzy rules involving fewer than 2 antecedent conditions. Following these two, C45-IFRC also enjoys high structural interpretability, being able to learn rules of the third shortest on average in length. UR-IFRC also learns short fuzzy rules employing only fewer than 4 antecedent variables on average. In contrast, MOGUL and QSBA have a fixed length for each fuzzy rule that is set according to the problem dimensionality and hence, their returned rules are typically rather complex. As for GP-COACH and SLAVE2, *Cond* only counts the number of the antecedent variables appearing in the rule, not the additional complexity incurred due to their use of compounded fuzzy terms in describing the variables.

For *Rul*, PTTD and QSBA return rule bases with the smallest size, due to their imposed heuristic nature of setting the number of rules to the number of the classes. However, the interpretability of QSBA model is poor since the rules it returns are very complicated, involving all variables for each rule. In general, both PTTD and SGERD tend to generate most compact rule bases with not only very small rule sizes but also short rules. Yet, their classification performances are poor compared with that of the proposed approach. One possible reason that UR-IFRC learns rule bases with a larger size may be due to the fact that UR generates rules with more antecedent variables, which is caused by its specific parameter (maxDepth) setting as indicated in Table 3. Importantly, C45-IFRC is able to learn rule bases of a small cardinality (each time returning fewer than 12 rules required on average across the 16 data sets), simpler than those returned by GP-COACH, MOGUL, and FH-GBML.

4.5. Comparison with non-fuzzy-rule-based classifiers

In addition to comparing against alternative fuzzy rule-based learning classifiers, the performance of the proposed approach is further compared with another 6 popular learning classifiers which are non-fuzzy-rule-based. Table 9 summarises the classification accuracy and Table 10 shows the *t*-test results. The six compared methods are:

SMO [36] is a sequential optimisation algorithm for building support vector machines (which form another type of most popular learning classifiers), with the polynomial kernel adopted as kernel function.

IBk [37] is the classical *k*-nearest neighbour approach, where an instance is classified by a majority vote of its neighbours. It works by assigning an instance to the class most common among its *k* nearest neighbours.

FRNN [38] is a fuzzy-rough set-based nearest neighbour classification algorithm, which uses the nearest neighbours to construct lower and upper approximations of decision classes, and classifies instances based on their membership to these approximations.

NB [39] or Naive Bayes is a simple probabilistic learning classifier, based on direct application of the Bayesian theorem with strong independence assumptions.

RIPPER [40] is a crisp rule induction algorithm following a separate-and-conquer strategy. Crisp rules are created incrementally one at a time, followed by an immediate simplification procedure. Once a set of rules for a given class is completed, an optimisation process is further imposed to fine-tune the rules.

NFC [41] is an improved version of the powerful adaptive-network-based fuzzy inference system ANFIS [42]. NFC improves

Table 9
Comparison of classification accuracy (%) against non-fuzzy-rule-based classifiers.

Data Set	C45-IFRC	UR-IFRC	RIPPER	NB	SMO	IBk	FRNN	NFC
appendicitis	84.34 ± 2.63	86.83 ± 1.70	79.51 ± 2.61	85.21 ± 0.52	**87.42 ± 0.77**	80.96 ± 1.22	83.72 ± 1.42	85.56 ± 1.75
banknote	98.63 ± 0.34	98.75 ± 0.22	98.41 ± 0.32	84.01 ± 0.14	97.97 ± 0.07	99.85 ± 0.00	99.85 ± 0.00	**99.94 ± 0.05**
blood	77.53 ± 0.48	77.82 ± 1.11	69.60 ± 0.97	75.28 ± 0.37	76.18 ± 0.09	71.06 ± 0.76	71.50 ± 0.85	**78.46 ± 0.48**
breast-cancer	95.15 ± 0.68	95.90 ± 0.39	93.71 ± 1.08	96.12 ± 0.08	**96.77 ± 0.18**	95.35 ± 0.24	96.45 ± 0.19	95.68 ± 0.20
column-2c	80.16 ± 2.16	81.00 ± 2.07	76.71 ± 0.81	77.87 ± 0.27	78.90 ± 0.81	81.06 ± 1.19	78.68 ± 1.26	**85.00 ± 0.57**
column-3c	77.51 ± 1.90	78.76 ± 1.68	76.81 ± 2.43	82.58 ± 0.59	76.10 ± 0.83	76.74 ± 0.93	75.68 ± 0.84	**84.23 ± 0.94**
ionosphere	86.80 ± 0.72	85.58 ± 1.84	84.04 ± 2.13	83.78 ± 0.21	82.96 ± 0.76	85.17 ± 0.78	**89.22 ± 0.45**	85.62 ± 1.78
iris	95.32 ± 0.54	95.59 ± 0.56	94.40 ± 1.61	95.53 ± 0.45	**96.27 ± 0.47**	95.40 ± 0.38	94.07 ± 0.38	93.80 ± 1.81
liver-disorders	64.83 ± 1.64	64.86 ± 2.09	62.35 ± 2.86	54.89 ± 1.14	57.98 ± 0.24	62.22 ± 1.15	62.81 ± 0.97	**70.35 ± 1.17**
mammographic	79.13 ± 0.79	78.46 ± 1.09	78.96 ± 1.05	77.64 ± 0.53	79.39 ± 0.34	74.91 ± 0.60	74.11 ± 0.56	**81.56 ± 0.37**
new-thyroid	91.88 ± 1.20	94.29 ± 0.85	88.99 ± 1.66	96.92 ± 0.25	89.30 ± 0.51	96.93 ± 0.57	**97.39 ± 0.79**	95.27 ± 2.25
parkinsons	84.33 ± 1.11	87.02 ± 1.34	88.24 ± 1.99	70.14 ± 0.59	87.00 ± 0.61	**95.90 ± 0.41**	93.96 ± 0.46	83.23 ± 0.45
pima-diabetes	75.05 ± 0.89	75.27 ± 0.69	66.88 ± 1.62	75.76 ± 0.44	**76.80 ± 0.24**	70.62 ± 0.84	69.07 ± 0.89	75.68 ± 0.86
seeds	91.37 ± 1.25	92.66 ± 1.52	87.52 ± 1.25	90.53 ± 0.47	93.57 ± 0.34	**93.86 ± 0.76**	93.09 ± 0.82	91.14 ± 1.17
sonar	72.59 ± 4.21	77.39 ± 1.98	73.92 ± 2.39	67.71 ± 1.08	76.59 ± 1.94	**86.17 ± 0.84**	85.25 ± 0.62	76.00 ± 1.73
wdbc	94.30 ± 0.53	94.98 ± 0.73	93.82 ± 0.80	93.31 ± 0.17	**97.54 ± 0.22**	95.64 ± 0.28	95.29 ± 0.39	94.52 ± 0.57
Summary	84.308	85.323	82.117	81.704	84.421	85.116	85.009	**86.003**

Table 10
Comparison against non-fuzzy-rule-based classifiers, where v, -, and * indicate statistically better, same, and worse classification performance against the proposed approach.

Data sets	C45-IFRC						UR-IFRC					
	RIPPER	NB	SMO	IBk	FRNN	NFC	RIPPER	NB	SMO	IBk	FRNN	NFC
appendicitis	*	-	v	*	-	-	*	*	-	*	*	*
banknote-authentication	-	*	*	v	v	v	*	*	*	v	v	v
blood	*	*	*	*	*	v	*	*	*	*	*	-
breast-cancer-wisconsin	*	v	v	-	v	v	*	-	v	*	v	-
column-2C	*	*	-	-	*	v	*	*	*	-	*	v
column-3C	-	v	*	-	*	v	*	v	*	*	*	v
ionosphere	*	*	*	*	v	*	*	*	*	-	v	-
iris	*	-	v	-	*	*	*	-	v	-	*	*
liver-disorders	*	*	*	*	*	v	*	*	*	*	*	v
mammographic	-	*	-	*	*	v	-	*	v	*	*	v
new-thyroid	*	v	*	v	v	v	*	v	*	v	v	-
parkinsons	v	*	v	v	v	*	-	*	-	v	v	*
pima-diabetes	*	v	v	*	*	-	*	-	v	*	*	-
seeds	*	*	v	v	v	-	*	*	v	v	-	*
sonar	-	*	v	v	v	v	*	*	-	v	v	-
wdbc	*	*	v	v	v	-	*	*	v	v	-	*
Summary (*/-/v)	(11/4/1)	(10/2/4)	(6/2/8)	(6/4/6)	(7/1/8)	(3/4/9)	(14/2/0)	(11/3/2)	(7/3/6)	(7/3/6)	(8/2/6)	(5/6/5)

ANFIS by adopting an advanced optimisation algorithm to help refine system parameters.

Compared with NB and RIPPER, UR-IFRC performs significantly better, in terms of both the average accuracy and the *t*-test results. Such clear wins are also achieved by C45-IFRC, compared to NB and RIPPER. The gap regarding the accuracy between C45-IFRC and the well-designed and robust SVM classifier is only fewer than 0.1%. Statistically, the results are also close to those of running SVM and nearest neighbour-based learning classifiers. Whereas UR-IFRC does not outperform the rest for any data set, it achieves better averaged accuracy than SMO, IBK and FRNN, supported with better statistical results, and a statistically equal performance with NFC. Collectively, the resultant fuzzy rule bases have demonstrated a promising performance that is at least comparable to the popular, well-established non-fuzzy-rule-based classifiers. Importantly, such an excellent performance is achieved using only fixed quantity space with interpretable inference results, forming a sharp contrast with SVM and nearest neighbour-based learning classifiers (whose results are generally difficult to interpret).

4.6. Effect of local rule selection

The above experimental results have demonstrated the promising performance of the proposed approach, in terms of both classification accuracy and model interpretability (thanks to the use of only predefined fuzzy sets and the induction of compact rules and rulesets). The high comprehensibility is achieved without embedding any sophisticated criterion in the final GA-based tuning step (by setting $w_i = 0$ as indicated in Table 3). However, such compact and transparent rule bases cannot be obtained without the stage of local rule selection through functional generalisation. This is confirmed with the further experimental investigations as reported below.

In conducting this purposefully devised experimentation, 3 different assignments for the interpretability weight w_i, as given in Eq. (22) are used, namely: 0.0, 0.1 and 1.0. Without overly complicating the experimentation, a single run based on 10-fold cross validation (10-CV) is performed for 5 data sets with C45-IFRC. Results are averaged, and analysed, in terms of: training accuracy (*Trn*), testing accuracy (*Tst*), average number of rules (R_1) after Stage 1 (i.e., the average number of potential fuzzy rules after heuristic mapping procedure), average number of rules (R_2) after Stage 2 (i.e., the number of all returned fuzzy rules with the local rule selection procedure), and average number of rules (R_3) after Stage 3 (i.e., the size of the final ruleset). The average number of antecedent variables, or the conditions (*Cond*), per resultant rule is also recorded together with the execution time (*Time*) for each complete 10-CV run.

As shown in Table 11, the reduction in the number of rules obtained after local rule selection is significant, R_2 is at least 10 times smaller than R_1 (for the data set column-3C, it is over 20

Table 11
Analysis of local rule selection.

Data sets	Setup	Trn	Tst	R_1	R_2	R_3	Cond	Time
column-2C	with stage 2, $w_i = 0$	81.5	77.1	101.1	7.5	5.8	2.2	29.4
	without stage 2, $w_i = 0$	75.9	71.6	101.1	101.1	48.4	2.7	109.1
	without stage 2, $w_i = 0.1$	74.4	70.3	101.1	101.1	16.6	2.7	35.7
	without stage 2, $w_i = 1$	64.8	61.6	101.1	101.1	1.6	1.0	3.4
column-3C	with stage 2, $w_i = 0$	78.5	76.5	141.6	6.0	4.2	2.0	34.1
	without stage 2, $w_i = 0$	54.0	50.7	141.6	141.6	30.2	2.5	52.4
	without stage 2, $w_i = 0.1$	55.6	56.8	141.6	141.6	11.7	2.2	24.3
	without stage 2, $w_i = 1$	46.8	50.3	141.6	141.6	3.8	1.4	7.0
ionosphere	with stage 2, $w_i = 0$	89.5	85.2	93.0	8.5	8.1	2.3	15.4
	without stage 2, $w_i = 0$	88.8	82.6	93.0	93.0	79.1	2.8	93.1
	without stage 2, $w_i = 0.1$	89.8	84.4	93.0	93.0	32.4	2.6	62.6
	without stage 2, $w_i = 1$	72.6	73.5	93.0	93.0	6.8	1.5	18.5
seeds	with stage 2, $w_i = 0$	93.5	91.0	85.8	8.0	7.3	2.1	18.0
	without stage 2, $w_i = 0$	91.5	89.0	85.8	85.8	56.1	2.6	60.3
	without stage 2, $w_i = 0.1$	92.0	90.0	85.8	85.8	21.9	2.2	27.7
	without stage 2, $w_i = 1$	57.5	55.2	85.8	85.8	5.3	1.7	5.7
wdbc	with stage 2, $w_i = 0$	95.3	94.2	133.2	12.8	9.7	2.5	78.1
	without stage 2, $w_i = 0$	95.1	93.3	133.2	133.2	100.6	2.8	258.1
	without stage 2, $w_i = 0.1$	95.0	93.0	133.2	133.2	23.2	2.5	110.9
	without stage 2, $w_i = 1$	77.7	78.8	133.2	133.2	5.6	1.6	24.7

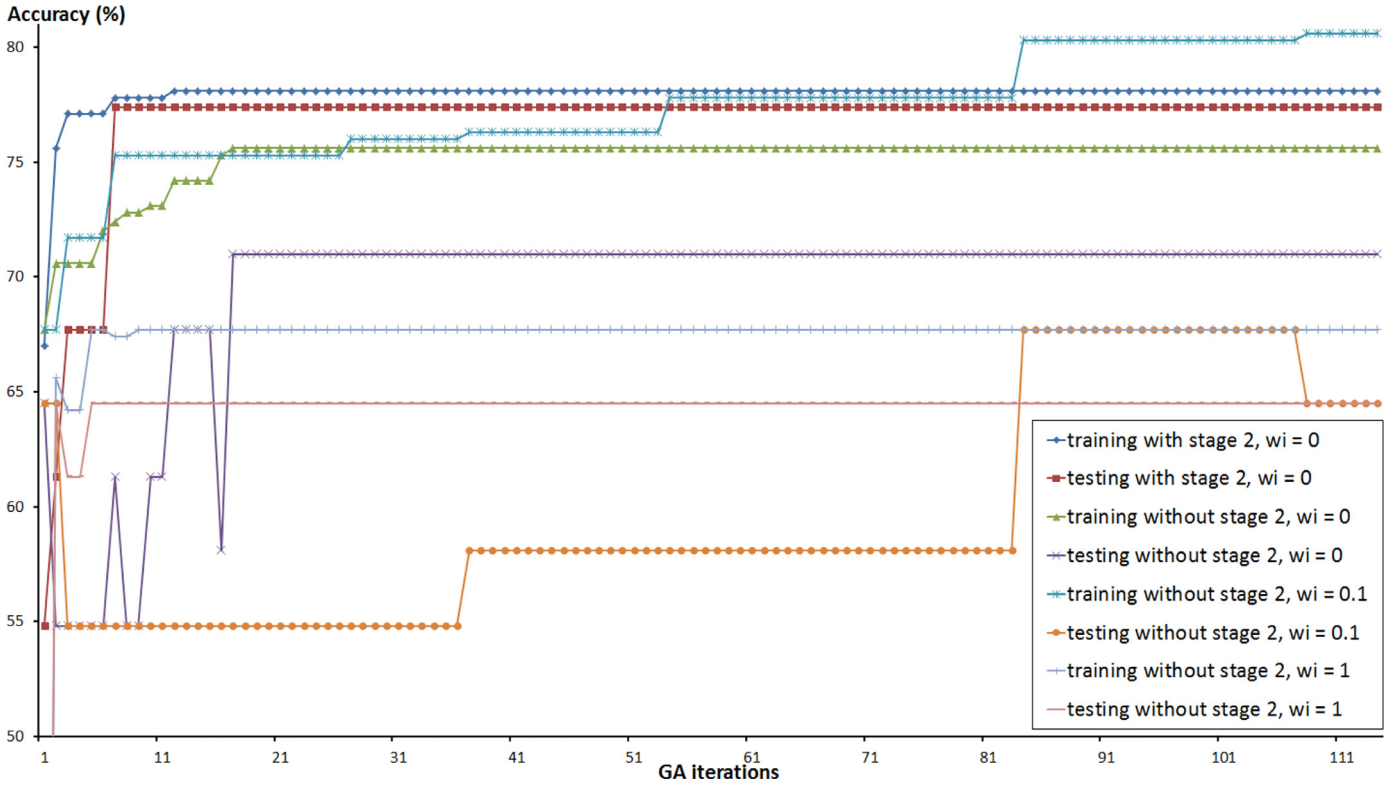

Fig. 4. Example genetic tuning runs (on the data set column-2C).

times smaller). Such reduction still results in a highly compact rule base, even when the interpretability weight is not included in the subsequent genetic tuning. The size of the resultant rule base after running Stage 2 is generally over 2 times smaller than that returned without the stage when $w_i = 0.1$, and much less when $w_i = 0$ (more than 10 times).

Recall that such substantial reduction is obtained with a design that is subject to functional generalisation, without loss in the performance of selected rules. However, if Stage 2 is not run, when $w_i = 1$, although a very small rule base with short rules may be returned, the classification performance is significantly decreased. Better performances are generally achieved when $w_i = 0$ or $w_i = 0.1$ in terms of accuracy, yet all of which are still worse than those with Stage 2 being run. In particular, regarding the data sets column-2C and column-3C, the resultant classification accuracies without Stage 2 are far worse than those achievable with the local selection procedure turned on. As an example, Fig. 4 shows a single GA run (with the parameter settings as given in the figure), regarding both training and testing accuracy. When running with Stage 2 it only takes a few generations to converge. In situations where Stage 2 is not implemented, the plot on the testing accuracy oscillates before it settles down around 20th generation when $w_i = 0$; it takes more than 100 generations to converge in case of $w_i = 0.1$; whereas when interpretability is weighted significantly higher, GA even fails to find solutions with good performance.

Running the local rule section procedure requires additional computation in search, where GA needs to run multiple times (with the number depending on that of the given crisp rules) as overheads. However, in real applications of the proposed approach, such multiple search attempts can be realised in parallel in order to reduce the otherwise required time for series implementation. Despite the time measured in this experiment is obtained by running multiple GAs sequentially, the result is very promising as such additional cost helps reduce the overall run time that the final genetic tuning will spend. As demonstrated by the results, the overall run time cost is generally much smaller than that is required without running the local rule selection (when $w_i = 0$ or $w_i = 0.1$).

5. Conclusion

Owing to the significance of incorporating consistent domain expertise by the use of predefined fuzzy sets, this paper has proposed a novel approach to generating interpretable fuzzy classification rules. For a given classification problem, simple crisp rules are utilised for initialisation, with each of them pointing to the model sub-spaces where desirable fuzzy rules potentially exist. This is followed by a heuristic mapping procedure that converts each preliminary crisp rule into a set of interpretable fuzzy rules involving only the predefined fuzzy sets, ensuring semantic interpretability. A local rule selection procedure is then performed to obtain a compact subset of initially mapped fuzzy rules that jointly generalise the capability of the underlying crisp rule. A fine grain tuning of all selected subsets of fuzzy rules is finally carried out with a conventional GA, resulting in an accurate and interpretable fuzzy ruleset to support the building of a KBS for pattern classification with a simplified structure.

Systematic experimental examinations of the proposed approach have been carried out, involving the use of two different crisp rule generation mechanisms for initialisation, over 16 benchmark datasets, in comparison with 7 alternative fuzzy learning classifiers and 6 popular non-fuzzy-rule-based classifiers. The results have revealed the general superiority of the proposed approach over alternative interpretable fuzzy classifiers employing only fixed and predefined fuzzy sets. They are also competitive to sophisticated data-driven non-fuzzy-rule-based state-of-the-art methods whose results are not directly interpretable. Indeed, the introduced functional generalisation method has proven effective in the production of the fuzzy rule bases, which are of high interpretability, being compact with short rules and exhibiting semantic comprehensibility.

In the present implementation, multiple modelling objectives are simply converted into a compound single objective using weights. However, it would be interesting to investigate whether the problem could be directly tackled using multi-objective evolutionary algorithms [30], enabling different tradeoffs between the possibly competing objectives. Also, the optimisation is currently realised with a genetic algorithm which is satisfactory but the underlying approach is more general and can be implemented with other techniques. Another piece of further research would therefore, be to explore the possibility of replacing the GA with alternative population-based algorithms such as harmony search [43].

References

[1] M.J. Cobo, M. Martínez, M. Gutiérrez-Salcedo, H. Fujita, E. Herrera-Viedma, 25 Years at knowledge-based systems: a bibliometric analysis, Knowl. Based Syst. 80 (2015) 3–13.

[2] M. Nakano, A. Takahashi, S. Takahashi, Fuzzy logic-based portfolio selection with particle filtering and anomaly detection, Knowl. Based Syst. 131 (2017) 113–124.

[3] M. Nilashi, R. Zakaria, O. Ibrahim, M.Z.A. Majid, R.M. Zin, M.W. Chugtai, N.I.Z. Abidin, S.R. Sahamir, D.A. Yakubu, A knowledge-based expert system for assessing the performance level of green buildings, Knowl. Based Syst. 86 (2015) 194–209.

[4] P. Su, Q. Shen, T. Chen, C. Shang, Ordered weighted aggregation of fuzzy similarity relations and its application to detecting water treatment plant malfunction, Eng. Appl. Artif. Intell. 66 (2017) 17–29.

[5] A. Segatori, F. Marcelloni, W. Pedrycz, On distributed fuzzy decision trees for big data, IEEE Trans. Fuzzy Syst. 26 (1) (2018) 174–192.

[6] A.M. Palacios, J.L. Palacios, L. Sánchez, J. Alcalá-Fdez, Genetic learning of the membership functions for mining fuzzy association rules from low quality data, Inf. Sci. 295 (2015) 358–378.

[7] R. Senge, E. Hüllermeier, Fast fuzzy pattern tree learning for classification, IEEE Trans. Fuzzy Syst. 23 (6) (2015) 2024–2033.

[8] T. Chen, Q. Shen, P. Su, C. Shang, Fuzzy rule weight modification with particle swarm optimisation, Soft Comput. 20 (8) (2016) 2923–2937.

[9] K. Cpałka, Design of interpretable fuzzy systems, Stud. Comput. Intell. 684 (1) (2017).

[10] J.M. Alonso, C. Castiello, C. Mencar, Interpretability of fuzzy systems: current research trends and prospects, in: Springer Handbook of Computational Intelligence, Springer, 2015, pp. 219–237.

[11] C. Mencar, A.M. Fanelli, Interpretability constraints for fuzzy information granulation, Inf. Sci. 178 (24) (2008) 4585–4618.

[12] S.-M. Zhou, J.Q. Gan, Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling, Fuzzy Sets Syst. 159 (23) (2008) 3091–3131.

[13] A. Lotfi, H.C. Andersen, A.C. Tsoi, Interpretation preservation of adaptive fuzzy inference systems, Int. J. Approx. Reason. 15 (4) (1996) 379–394.

[14] J.G. Marín-Blázquez, Q. Shen, From approximative to descriptive fuzzy classifiers, IEEE Trans. Fuzzy Syst. 10 (4) (2002) 484–497.

[15] D. Soria, J.M. Garibaldi, A.R. Green, D.G. Powe, C.C. Nolan, C. Lemetre, G.R. Ball, I.O. Ellis, A quantifier-based fuzzy classification system for breast cancer patients, Artif. Intell. Med. 58 (3) (2013) 175–184.

[16] N. Nithya, K. Duraiswamy, Correlated gain ratio based fuzzy weighted association rule mining classifier for diagnosis health care data, J. Intell. Fuzzy Syst. 29 (4) (2015) 1453–1464.

[17] S. Mabu, C. Chen, N. Lu, K. Shimada, K. Hirasawa, An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 41 (1) (2011) 130–139.

[18] P. Su, C. Shang, T. Chen, Q. Shen, Exploiting data reliability and fuzzy clustering for journal ranking, IEEE Trans. Fuzzy Syst. 25 (5) (2017) 1306–1319.

[19] J. Wyatt, Nervous about artificial neural networks? The Lancet 346 (8984) (1995) 1175–1177.

[20] R. Alcalá, Y. Nojima, F. Herrera, H. Ishibuchi, 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) (2011) 2303–2318.

[21] H. Ishibuchi, S. Mihara, Y. Nojima, Parallel distributed hybrid fuzzy gbml models with rule set migration and training data rotation, IEEE Trans. Fuzzy Syst. 21 (2) (2013) 355–368.

[22] H. Ishibuchi, T. Yamamoto, T. Nakashima, Hybridization of fuzzy GBML approaches for pattern classification problems, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 35 (2) (2005) 359–365.

[23] D. García, A. González, R. Pérez, Overview of the slave learning algorithm: a review of its evolution and prospects, Int. J. Comput. Intell. Syst. 7 (6) (2014) 1194–1221.

[24] A. González, R. Pérez, Selection of relevant features in a fuzzy genetic learning algorithm, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 31 (3) (2001) 417–425.

[25] R. Senge, E. Hullermeier, Top-down induction of fuzzy pattern trees, IEEE Trans. Fuzzy Syst. 19 (2) (2011) 241–252.

[26] F.J. Berlanga, A. Rivera, M.J. del Jesús, F. Herrera, Gp-coach: genetic programming-based learning of compact and accurate fuzzy rule-based classification systems for high-dimensional problems, Inf. Sci. 180 (8) (2010) 1183–1200.

[27] E.G. Mansoori, M.J. Zolghadri, S.D. Katebi, Sgerd: a steady-state genetic algorithm for extracting fuzzy classification rules from data, IEEE Trans. Fuzzy Syst. 16 (4) (2008) 1061–1071.

[28] P. Pulkkinen, H. Koivisto, A dynamically constrained multiobjective genetic fuzzy system for regression problems, IEEE Trans. Fuzzy Syst. 18 (1) (2010) 161–177.

[29] J. Hühn, E. Hüllermeier, Furia: an algorithm for unordered fuzzy rule induction, Data Min. Knowl. Discov. 19 (3) (2009) 293–319.

[30] A. Fernández, V. López, M.J. del Jesus, F. Herrera, Revisiting evolutionary fuzzy systems: taxonomy, applications, new trends and challenges, Knowl. Based Syst. 80 (2015) 109–121.

[31] K. Bache, M. Lichman, UCI Machine Learning Repository, 2013.

[32] I.H. Witten, E. Frank, M.A. Hall, C.J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2016.

[33] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Mult.-Valued Logic Soft Comput. 17 (2–3) (2010) 255–287.

[34] O. Cordón, M.J. del Jesús, F. Herrera, M. Lozano, Mogul: a methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach, Int. J. Intell. Syst. 14 (11) (1999) 1123–1153.

[35] K.A. Rasmani, Q. Shen, Data-driven fuzzy rule generation and its application for student academic performance evaluation, Appl. Intell. 25 (3) (2006) 305–319.

[36] J. Shawe-Taylor, S. Sun, A review of optimization methodologies in support vector machines, Neurocomputing 74 (17) (2011) 3609–3618.

[37] I. Triguero, J. Derrac, S. Garcia, F. Herrera, A taxonomy and experimental study on prototype generation for nearest neighbor classification, IEEE Tran. Syst. Man Cybern. Part C: Appl. Rev. 42 (1) (2012) 86–100.
[38] R. Jensen, C. Cornelis, Fuzzy-rough nearest neighbour classification, in: Transactions on Rough Sets XIII, Springer, 2011, pp. 56–72.
[39] K.P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.
[40] W.W. Cohen, Fast effective rule induction, in: Proceedings of the Twelfth International Conference on Machine Learning, 1995, pp. 115–123.
[41] B. Cetişli, A. Barkana, Speeding up the scaled conjugate gradient algorithm and its application in neuro-fuzzy classifier training, Soft Comput. 14 (4) (2010) 365–378.
[42] J.-S. Jang, Anfis: adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man Cybern. 23 (3) (1993) 665–685.
[43] R. Diao, Q. Shen, Feature selection with harmony search, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 42 (6) (2012) 1509–1523.