Information Sciences 180 (2010) 1183–1200

Contents lists available at ScienceDirect

Information Sciences

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

GP-COACH: Genetic Programming-based learning of COmpact and ACcurate fuzzy rule-based classification systems for High-dimensional problems ☆

F.J. Berlanga [a,*], A.J. Rivera [b], M.J. del Jesus [b], F. Herrera [c]

[a] *University of Zaragoza, Dept. of Computer Science and Systems Engineering, E-50018 Zaragoza, Spain*
[b] *University of Jaén, Dept. of Computer Science, E-23071 Jaén, Spain*
[c] *University of Granada, Dept. of Computer Science and Artificial Intelligence, E-18071 Granada, Spain*

ARTICLE INFO

Article history:
Received 7 October 2008
Received in revised form 25 November 2009
Accepted 22 December 2009

Keywords:
Classification
Genetic programming
Fuzzy rule-based systems
Genetic fuzzy systems
High-dimensional problems
Interpretability-accuracy trade-off

ABSTRACT

In this paper we propose GP-COACH, a Genetic Programming-based method for the learning of COmpact and ACcurate fuzzy rule-based classification systems for High-dimensional problems. GP-COACH learns disjunctive normal form rules (generated by means of a context-free grammar) coded as one rule per tree. The population constitutes the rule base, so it is a genetic cooperative-competitive learning approach. 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 and allows the obtaining of a compact set of fuzzy rules. The results obtained have been validated by the use of non-parametric statistical tests, showing a good performance in terms of accuracy and interpretability.

1. Introduction

In the design of any fuzzy rule-based system (FRBS) learning method, there are two main (and contrary) goals to be maximized: the accuracy and the interpretability of the knowledge extracted. In the 1990s, more attention was given to accuracy maximization, and different approaches were developed to improve the accuracy of FRBSs, although this improvement was usually at the cost of their interpretability. However, more recent studies [17–19,24,41,46] have indicated the necessity of an interpretability-accuracy trade-off in the design of FRBSs.

Such a trade-off is more difficult to achieve when the problem to be solved has high dimensionality, that is, a high number of input features or a high number of examples. In this paper we consider the high dimensionality problem with regard to the number of features. In this kind of problems, a linear increase in the number of input features causes an exponential growth of the fuzzy rule search space, what is popularly known as the combinatorial rule explosion problem [23]. This growth makes the learning process more difficult and, in most cases, leads to an FRBS with a high level of complexity (with respect to the number of rules, features and conditions included in each rule).

An analysis of the specialized literature shows that there exist two main solutions for tackling this problem of high dimensionality in the learning of compact, interpretable and accurate FRBSs:

☆ Supported by the Spanish Ministry of Science and Technology under the Projects TIN2008-06681-C06-02 and TIN2008-06681-C06-01 and the Andalusian Research Plan TIC-3928.

* Corresponding author.
E-mail addresses: berlanga@unizar.es (F.J. Berlanga), arivera@ujaen.es (A.J. Rivera), mjjesus@ujaen.es (M.J. del Jesus), herrera@decsai.ugr.es (F. Herrera).

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2009.12.020

1. *Carrying out a feature selection process* [40,50,54], which determines the most relevant variables before (*a priori feature selection*) or during (*embedded feature selection*) the FRBS inductive learning process. This process reduces the fuzzy rule search space and increases the efficiency and accuracy of the learning stage.
2. *Compacting and reducing a previously learned rule set in a postprocessing stage* [16,48,65]. The methods which employ this strategy operate by combining rules and/or selecting a subset of them from a given rule set in order to achieve the goal of minimizing the number of rules used while maintaining (or even improving) the FRBS performance.

Evolutionary algorithms (EAs), and particularly genetic algorithms (GAs) [36], have been successfully applied to FRBS learning, giving way to the appearance of the so-called genetic fuzzy systems (GFSs) [27,41,46]. Many different GFSs have been formulated for the learning of fuzzy rule sets. Although most of these methods are GA-based, it is also possible to find proposals using other different types of EAs such as genetic programming (GP) [53], a type of evolutionary algorithm that uses variable-length trees to represent the different individuals in the population, instead of fixed-sized vectors with binary, integer or real codification [35,56,62,68].

In this paper we propose GP-COACH (Genetic Programming-based learning of COmpact and ACcurate fuzzy rule-based classification systems for High-dimensional problems), a method for dealing with problems having a high dimensionality with regard to the number of input features considered. It is a GP-based method that allows the absence of some of the input features in each rule. GP-COACH learns disjunctive normal form (DNF) fuzzy rules (generated by means of a context-free grammar), coded as one rule per chromosome and with the population forming the rule set, thus following a genetic cooperative-competitive learning approach [41]. It uses a competition mechanism between rules (token competition) which simultaneously maintains the diversity in the population and deletes irrelevant rules during the learning process, allowing us to obtain compact FRBCSs (with few rules, variables, and conditions per rule) with a high generalization capability.

An experimental study involving 24 data sets and five well-known FRBCS learning algorithms has been carried out. Nonparametric statistical methods have been used to compare and analyze the compactness and accuracy of the experimental results. They show the good performance (in terms of accuracy and interpretability) of our approach. Moreover, the suitability of some GP-COACH components such as the token competition diversity mechanism, the use of specific genetic operators and the advantages of using a GP algorithm instead of a traditional GA have been also analyzed.

This paper is organized in the following way. In Section 2, some preliminaries are described. The GP-COACH algorithm is comprehensively described in Section 3. The experimental framework is presented in Section 4. In Sections 5 and 6 we have included the experimental results and their analysis. Finally, in Section 7 some concluding remarks are pointed out.

2. Preliminaries

In this section, we introduce the notation that has been used in this paper. Then we describe the structure of an FRBCS, a brief introduction to GFSs and finally a short review of the main approaches we find in the specialized literature on the use of GP for learning FRBSs.

2.1. Notation

A classification problem is considered with:

- A set of input variables $X = \{X_i / i = 1, \ldots, n_v\}$, where n_v is the number of features of the problem.
- A set of values for the target variable (class) $C = \{C^j / j = 1, \ldots, n_c\}$, where n_c is the number of different values for the class variable.
- A set of examples $E = \{e^h = (e^h_1, \ldots, e^h_{n_v}, C^h) / h = 1, \ldots, n_e\}$, where C^h is the class label for the sample e^h, and n_e is the number of examples.

2.2. Fuzzy rule-based classification systems

FRBCSs have been successfully applied to pattern classification problems [9,46], and the interest in their use arises from the fact that they provide a good platform for managing noisy, imprecise or incomplete information, which is often encountered in any human-cognition system.

An FRBCS is composed of a knowledge base and a fuzzy reasoning method. Both components are described in the next subsections.

2.2.1. The knowledge base

Composed of a rule base (RB) and a data base (DB):

- *Rule base*. Our method learns RBs containing the following type of rules:

$$R^k : \text{If } X_1 \text{ is } \widehat{A}_1^k \text{ and } \ldots \text{ and } X_{n_v} \text{ is } \widehat{A}_{n_v}^k \text{ then } \mathit{Class} \text{ is } C^k \text{ with } CD^k, \quad (1)$$

where each input variable X_i takes as a value a set of linguistic labels $\widehat{A}_i^k = \{L_i^1 \text{ or } \ldots \text{ or } L_i^{l_i}\}$ joined by a disjunctive operator, while the output variable (*Class*) is one of the class labels $C^k \in C$. This type of rule is called a DNF fuzzy rule and its structure uses a more compact description which improves the interpretability of the system. Moreover, the structure permits changes in granularity by means of the combination of linguistic terms using an *or* operation, and by its nature allows the absence of some input variables in each rule (simply letting $\widehat{A}_i^k$ be the whole set of linguistic terms). As can be seen, our DNF fuzzy rule also includes a certainty degree ($CD^k \in [0,1]$), which represents the confidence of the classification in the class label represented by the consequent of the rule (C^k). In our proposal, this certainty degree is obtained as follows:

$$CD^k = \frac{\sum\limits_{\substack{e^h \in C^k \\ h=1,\ldots,n_e}} m^k(e^h)}{\sum\limits_{h=1,\ldots,n_e} m^k(e^h)}, \quad (2)$$

where $m^k(e^h)$ is the degree of compatibility between an example and the antecedent part of a fuzzy rule, i.e., the degree of membership of the example to the fuzzy subspace delimited by the antecedent part of the rule, also known as the matching degree. In our method, this matching degree is defined as follows:

$$m_k(e^h) = T(TC(\mu_{L_1^1}(e_1^h), \ldots, \mu_{L_1^{l_1}}(e_1^h)), \ldots, TC(\mu_{L_{n_v}^1}(e_{n_v}^h), \ldots, \mu_{L_{n_v}^{l_{n_v}}}(e_{n_v}^h))), \quad (3)$$

where $L_i^{l_i}$ is the linguistic label number l_i of the variable i; $\mu_{L_i^{l_i}}(e_i^h)$ is the degree of membership for the value of the feature i of the example e^h to the fuzzy set corresponding to the linguistic label l_i for this variable (i); T is the t-norm selected to represent the meaning of the AND operator (the fuzzy intersection) that in our case is the minimum t-norm; and TC is the t-conorm selected to represent the meaning of the OR operator (the fuzzy union) that in our case is the maximum t-conorm. Other different methods for calculating this certainty degree can be found in [49].

- *Data base*. The DB contains the definition of the fuzzy sets related to the linguistic terms used in the RB. This fact leads us to specify the number of linguistic labels (l_i) for each variable X_i considered, and the membership function of the fuzzy sets related to these linguistic terms. In our experiments we have used five linguistic labels per variable, and we have divided each feature definition interval in a uniform manner using triangular fuzzy sets, which is a common way of defining the DB in the specialized literature. An example of this type of partition is shown in Fig. 1.

2.2.2. The fuzzy reasoning method

A fuzzy reasoning method (FRM) is an inference procedure that derives conclusions from a set of fuzzy if-then rules and a pattern. The power of fuzzy reasoning is that it is possible to achieve a result even when there is not an exact match (to a degree 1) between a system observation and the antecedents of the rules.

The most commonly used FRM, maximum matching, classifies an example using the rule consequent with the highest association degree, discarding the information given by other rules to a lesser degree. In [25], we presented a general reasoning model for combining information provided by different rules, which involves different possibilities as reasoning methods. In our experiments, we have used two different FRMs: the classical one and the normalized sum FRM [25,44].

2.3. Genetic fuzzy systems

A GFS is basically a fuzzy system augmented by a learning process based on evolutionary computation, which includes genetic algorithms (GAs), genetic programming (GP), and evolutionary strategies, among other evolutionary algorithms.

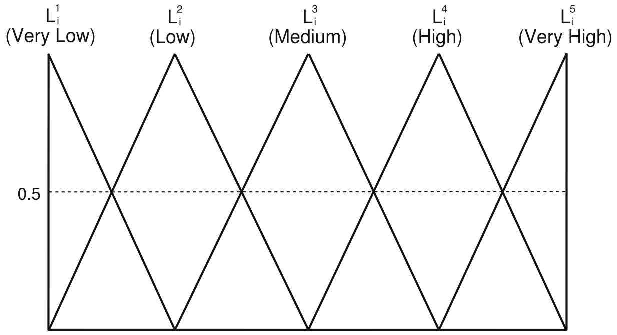

Fig. 1. A uniform fuzzy partition with triangular membership functions.

These well known and widely used global search techniques have the ability to explore a large search space for suitable solutions only requiring a performance measure. In addition to their ability to find near optimal solutions in complex search spaces, the generic code structure and independent performance features of EAs make them suitable candidates to incorporate a priori knowledge.

In the case of FRBSs, this a priori knowledge may be in the form of linguistic variables, fuzzy membership function parameters, fuzzy rules, number of rules, etc. These capabilities have extended the use of EAs in the development of a wide range of approaches for designing FRBSs over the last few years, as has been pointed out in the last special GFS issues [13,19,24,14].

In these special issues we can find studies emphasizing different research directions. Carse and Pipe's special issue [13] includes papers focused on the multiobjective evolutionary learning [37], boosting [63] and evolutionary adaptive inference systems [5]. Casillas et al.'s special issue [19] is focused on the trade-off between interpretability and accuracy, collecting papers that proposed different GFSs for tackling this problem: with multiobjective approaches [47,39], and optimizing the definition for the linguistic variables [4,11]. Cordón et al.'s [24] and Casillas and Carse's [14] special issues focus its attention on highly innovative GFS proposals that can mark new research trends: treatment of imprecise data in GFSs [61,64], incremental evolutionary learning [42], and multiobjective [10,33], parallel [57], genetic programming [56] and neuro-coevolutionary approaches [30].

In this paper, we will pay special attention to the GFSs which learn RBs. When considering the task of learning rules in a rule base based system, the different genetic learning methods follow two approaches in order to encode rules within a population of individuals:

- The "Chromosome = Set of rules", *Pittsburgh* approach, in which each individual represents a whole rule set [66,67].
- The "Chromosome = Rule" approach, in which each individual codifies a single rule, and the whole rule set is provided by combining several individuals in the population (rule cooperation) or via different evolutionary runs (rule competition). In turn, within the "Chromosome = Rule" approach, there are three generic proposals:
  - The *Michigan* approach (learning classifier system) [70,15], in which each individual codifies a single rule.
  - The *IRL* (*iterative rule learning*) approach [26,38], in which each chromosome represents a rule. The global solution is formed by the best rules obtained when the algorithm is run multiple times.
  - The *GCCL* (*genetic cooperative-competitive learning*) approach [45], in which the complete population or a subset of it codifies the RB. In this model, the chromosomes compete and cooperate simultaneously.

As was mentioned before, although most of the evolutionary proposals for the learning of GFSs are GA-based, it is also possible to find GFSs that use other different types of EAs. One example are those GFSs which use GP [53] for learning FRBSs. Most of these approaches follow the Pittsburgh codification scheme [1–3,35,43,55,60,62,68], although there are other proposals which use GCCL codification scheme [7,8,22,56].

An extensive review of the most recent developments of GFS and FRBS can be found in [41].[1]

3. GP-COACH algorithm

The main features of GP-COACH are the following:

- It uses a context-free grammar which allows the learning of DNF fuzzy rules (see Eq. (1)) and the absence of some input features, thus obtaining compact and simple rules.
- It follows the *GCCL* approach, so it encodes a single rule per individual and the RB is formed by the whole population. This makes it necessary to use two different fitness functions:
  - A local fitness function which evaluates the goodness of each of the different rules in the population of individuals. From now on, we will refer to it simply as *fitness function*.
  - A global fitness function which evaluates the goodness of a whole population of individuals (an RB). From now on, we will refer to it as *global fitness score*.
- It includes a mechanism to promote diversity in the population, in order to avoid the convergence of all individuals in the same area of search space. Specifically, it uses the *token competition* [71] diversity mechanism which makes rules compete among themselves during the evolutionary process, deleting irrelevant rules and thus generating a smaller number of rules that present a high generalization capability. We must emphasize that we have a variable size population, allowing us to obtain a small set of rules. In [51,52], studies on the use of variable size populations in GP are done.
- It uses a two level hierarchical inference process because it learns two different types of rules: *primary rules*, which are strong and general rules generated by the genetic operators, and *secondary rules*, which are weaker and more specific rules, generated after the token competition procedure in order to increase the diversity in the population.

[1] The web site http://www.sci2s.ugr.es/gfs/ provides complete information and material on the topic.

- GP-COACH uses a reproduction stage in which each child is created by applying only one of the genetic operators, and the children compete with their parents in order to generate a new population.

In the following subsections, these components, the complete description of the algorithm and a pseudo-code summarizing GP-COACH algorithm are shown.

3.1. Context-free grammar definition

DNF fuzzy rules are generated according to the production rules of a context-free grammar. In Table 1, an example of the grammar for a classification problem with two features (X_1, X_2), three linguistic labels per feature (*Low*,*Medium*,*High*) and three classes (C^1, C^2, C^3) is shown.

The symbol ?*a* in some of the production rules of the grammar represents one, and only one, of the values separated by commas in the square brackets.

3.2. Evaluating an individual: fitness function

Each one of the individuals (rules) in the population is evaluated according to a fitness function based on the estimation of two measurements: *Confidence*, which measures the accuracy of an individual, that is the confidence that the consequent will be true if the antecedent is verified (calculated in the same way as the certainty degree in Eq. (2)), and *Support*, which measures the extent of the knowledge represented in the individual:

$$Support(R^k) = \frac{\sum\limits_{\substack{e^h \in C^k \\ h=1,\dots,n_e}} m^k(e^h)}{N_{C^k}}, \tag{4}$$

where N_{C^k} is the number of examples having the same class as the one indicated in the consequent of the individual (R^k).

Both measurements are combined to form the fitness function in the following way:

$$raw\_fitness(R^k) = \alpha * Confidence(R^k) + (1 - \alpha) * Support(R^k), \tag{5}$$

where α parameter allows us to give more importance to either of these.

Finally, it is important to emphasize that each time that an individual is evaluated it is also necessary to modify its certainty degree according to its confidence value.

3.3. Evaluating a population: global fitness score

GP-COACH uses a *global fitness score* in order to obtain the best evolved population during the whole evolutionary process, defined as follows:

$$Global\_fitness = w_1 * accuracy + w_2 * (1.0 - Var_N) + w_3 * (1.0 - Cond_N) + w_4 * (1.0 - Rul_N), \tag{6}$$

where Var_N and $Cond_N$ are the normalized values of the average number of variables and conditions (labels) in the rules, and Rul_N the number of rules in the population, respectively. Table 2 shows the minimum and maximum values used to normalize each of these measurements.

It is important to point out that it is not possible to use a multiobjective evolutionary approach in GP-COACH for managing the different objectives due to our proposal uses the *GCCL* representation. We encode one fuzzy rule per chromosome and the complete solution is provided by joining all the individuals in the population, and therefore it is not possible to evolve a pareto of solutions. The design of a multiobjective evolutionary approach in GP will require the use of the Pittsburgh codification scheme, which presents several disadvantages such as efficiency problems, variable size codification problems, and code bloating that advice against its use for the obtaining of compact and accurate FRBCSs.

Table 1
Grammar example.

$Start \rightarrow [If], antec, [then], conseq, [\cdot]$
$antec \rightarrow descriptor1, [and], descriptor2$
$descriptor1 \rightarrow [any]$
$descriptor1 \rightarrow [X_1\ is]label$
$descriptor2 \rightarrow [any]$
$descriptor2 \rightarrow [X_2\ is]label.$
$label \rightarrow \{member(?a, [L, M, H, L\ or\ M, L\ or\ H, M\ or\ H, L\ or\ M\ or\ H])\}, [?a]$
$conseq \rightarrow [Class\ is]\ descriptorClass$
$descriptorClass \rightarrow \{member(?a, [C^1, C^2, C^3])\}, [?a]$

Table 2
Minimum and maximum values for normalization.

	Var	*Cond*	*Rul*
Min.	1	1	n_c
Max.	n_v	$n_v * (l_i - 1)$	n_e

3.4. Token competition: maintaining the diversity of the population

Token competition [71] has been used as a mechanism for maintaining the diversity in the population in GP-COACH. It emulates the following behavior in a natural environment: when an individual finds a good place to live (a niche) it will try to exploit this niche and prevent other newcomers from sharing its resources, unless a given newcomer is stronger than it is. The other individuals are hence forced to explore and find their own niches. In this way, the diversity of the population is increased.

Based on this idea, it is assumed that each example in the training set can provide a resource called a token, for which all chromosomes in the population will compete. If an individual (i.e. a rule) can match the example, it sets out a flag to indicate that the token is seized. Other weaker individuals then cannot capture the token.

The priority of receiving tokens is determined by the strength of the individuals. The individuals with a high fitness score will exploit their niches by seizing as many tokens as they can. The other individuals entering the same niches will have their strength decreased because they cannot compete with the stronger ones. This is achieved by introducing a penalization in the fitness score of each individual, based on the number of tokens which each individual has seized:

$$Penalized\_fitness(R^k) = \begin{cases} raw\_fitness(R^k) * \frac{count(R^k)}{ideal(R^k)}, & \text{if } ideal(R^k) > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where $raw\_fitness(R^k)$ is the fitness score obtained from the evaluation function, $count(R^k)$ is the number of tokens that the rule R^k actually seized and $ideal(R^k)$ is the total number of tokens that it can seize, which is equal to the number of examples that the rule matches.

As a result of token competition, there exist individuals that cannot seize any token. These individuals are considered as irrelevant, and they can be eliminated from the population due to the fact that all of their examples are covered by other stronger individuals.

In Fig. 2 a token competition example is shown. In this example there is a population with five rules $(R^1, \ldots, R^5)$ ordered decreasingly according to their fitness score. Before token competition is carried out, all the training examples $(e^1, \ldots, e^{12})$ have their tokens free. Once token competition has started, R^1 can seize all its tokens because it is the strongest rule. R^2 can only capture two of its three tokens because R^1 has previously seized the token associated with e^1, so R^2 must have its fitness score penalized. Rule R^3 must be eliminated from the population because all its tokens have been previously seized by other stronger rules, that is R^3 is considered as an irrelevant rule. R^4 and R^5 do not need to modify their fitness score because they can seize all their tokens.

The token competition let us eliminate rules which describe information provided by other stronger rules (with better confidence and/or support).

3.5. Secondary rules: improving population diversity

Once the token competition mechanism has been applied, it is possible that there exist training examples which have not been covered by any of the rules in the population (see examples e^4, e^5 and e^{11} in Fig. 2). The generation of new specific rules

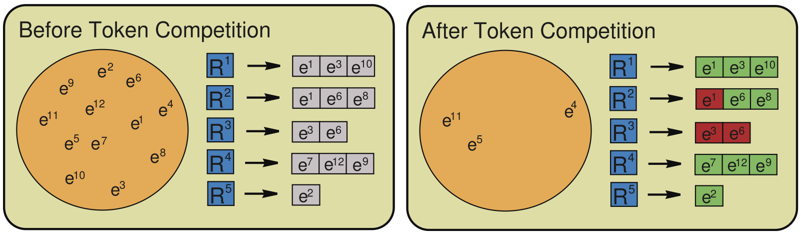

Fig. 2. Token competition example.

covering these examples improves the diversity of the population and helps the evolutionary process to find easily stronger and more general rules covering these examples.

As was previously mentioned, GP-COACH uses a two-level hierarchical inference process, learning rule sets with two different types of fuzzy rules: A core of strong and general rules (*primary rules*) which cover most of the examples, and a small set of weaker and more specific rules (*secondary rules*), which are only taken into account if there is no primary rule which matches the examples. This two-level hierarchical inference process works in the following way: When a new example *e* is given to a learned FRBCS, this will try to find the class that better suits *e* by using the information provided in its knowledge base. However, the FRM only will consider those rules which have been labelled as primary to obtain the class of the example *e*. The secondary rules will only be taken into account if there exists no primary rule in the RB matching the example *e*.

This two-level hierarchical inference process allows GP-COACH to improve the accuracy of the learned rule sets, avoiding misclassification errors coming from the use of the secondary rules when primary rules (more general ones) can be applied.

GP-COACH follows the method proposed by Chi et al. [21] (which extends the Wang and Mendel method [69] for classification problems) to generate *secondary rules*:

1. If after carrying out token competition, there are some of the training examples with their tokens free, then randomly choose one of them and generate a new rule covering this example. This new rule will contain all the input variables, and each one of these variables will only have a single linguistic label associated (the one for which the example presents the highest degree of membership). The class of the new rule will be the class associated with the example chosen.
2. All the examples that match the new rule, among those with their token free after token competition, are removed.
3. If there are still examples with their token free, then repeat the two previous steps. Otherwise, evaluate all the new rules that have been generated and join them to the ones in the current population.

It is important to point out that it is also possible that GP-COACH learns rule sets which have no secondary rules, because their primary rules are strong enough to cover all the given examples.

3.6. Genetic operators

GP-COACH makes use of four different genetic operators to generate new individuals (the operator selection procedure is described in the next subsection):

1. *Crossover* (Fig. 3): A part in the first parent is randomly selected and exchanged by another part, in the second one, but under the constraint that the offspring produced must be valid according to the grammar production rules. It is important to indicate that it is not possible to choose cut points in the middle of a disjunction of labels. This operator in fact produces two children, but only one of them (randomly chosen) is returned as a descendant.

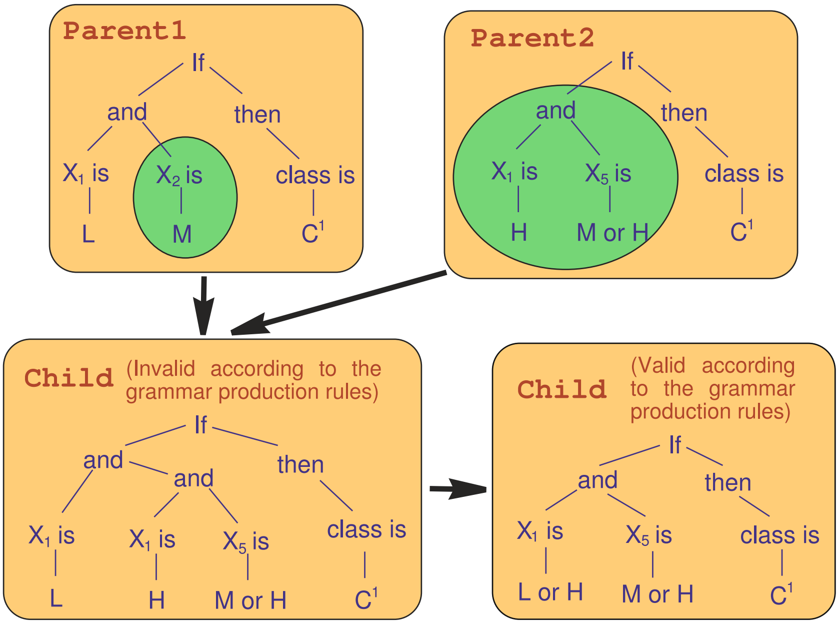

Fig. 3. Crossover operator.

2. *Mutation*: A variable in the rule is randomly chosen and then one of the following three actions is taken (again, randomly chosen):
   (a) A new label is added to the label set.
   (b) A label is removed from the label set.
   (c) A label is exchanged for another one not included in the label set.
3. *Insertion*: It inserts a new variable in the rule. The linguistic label set associated to this new variable is randomly chosen, although it must have at least one label and must be different from the "any" set (see Table 1).
4. *Dropping condition* (Fig. 4): Due to the probabilistic nature of GP, redundant constraints may be generated in the rule. Thus it is necessary to generalize the rules, to represent the knowledge in a more concise form. The dropping condition randomly selects one variable in the rule and then turns it into "any". The label set associated with this variable is also removed. The variable is no longer considered in the rule, hence the rule can be generalized.

3.7. Reproduction stage: selection, genetic operators application and replacement

GP-COACH generates a number of descendants equal to the size of the current population. This size can vary throughout the evolutionary process because of the action of token competition.

Selection: Parents are selected from the current population by the tournament selection scheme, with size 2.

Genetic operators application: Each individual chosen in the selection stage is used to generate a new child by applying one of the previously described four genetic operators. The genetic operators election is carried out in a probabilistic manner.

Two considerations need to be taken into account: The first one is related to the crossover operator. As has been said before, this operator generates one child from two parents. The tournament selection mechanism only returns one parent, so another parent must be selected from the current population. Nevertheless, in contrast with the first parent, this second one is randomly selected from all the individuals in the population. It must be noted that this crossover operator can swap information between individuals representing rules for different classes. It can increase the diversity in the population and if this swap generates rules with non-appropriate information our token competition mechanism will erase then from the population, keeping the good ones.

Another important issue is that it could be possible not to be able to apply all the genetic operators to a given parent. For instance, it is not possible to apply the dropping condition operator to a rule with a single variable or to apply the insertion operator to a rule using all the input variables. When this happens, another genetic operator is selected from the remaining ones.

Replacement: An important issue in GP-COACH is that the new population of children does not replace the current population of parents. Instead, a new joint population is formed by adding the parents and children populations. Individuals in this joint population are arranged by their fitness score in order to be able to apply the token competition diversity mechanism.

3.8. Description of the algorithm

The GP-COACH algorithm begins by creating an initial population according to the rules in the context-free grammar. Each individual in this population is then evaluated. Afterwards, the initial population is kept as the best-evolved population and its global fitness score is calculated. Then the initial population is copied to the current population and the evolutionary process begins:

1. An offspring population, with the same size as the current one, is created. Parents are selected by using the binary tournament selection mechanism, and children are created by using the genetic operators. Each child in this offspring population is then evaluated.

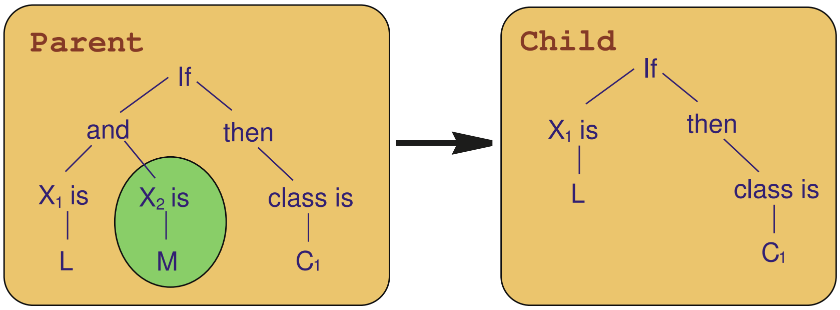

Fig. 4. Dropping condition operator.

```
Initialize(Initial_pop)
For each Initial_pop[i] do
   Initial_pop[i].fitness $\leftarrow$ Evaluate(Initial_pop[i])
End for
Copy Initial_pop to Best_pop
Best_pop.global_fitness $\leftarrow$ Global_score(Best_pop)
Copy Initial_pop to Current_pop
While (not termination-condition) do
    Offspring_pop = $\emptyset$
    While (Size(Offspring_pop) $\neq$ Size(Current_pop)) do
        Parent $\leftarrow$ Binary_tournament(Current_pop)
        Child $\leftarrow$ Genetic_operator(Parent)
        Evaluate(Child)
        Add Child to Offspring_pop
    End while
    Joint_pop $\leftarrow$ Current_pop $\cup$ Offspring_pop
    New_current_pop $\leftarrow$ Token_competition(Joint_pop)
    New_current_pop.global_fitness $\leftarrow$ Global_score(New_current_pop)
    If New_current_pop.global_fitness $>$ Best_pop.global_fitness then
        Best_pop.global_fitness $\leftarrow$ New_current_pop.global_fitness
        Copy New_current_pop to Best_pop
    End if
    Copy New_current_pop to Current_pop
End while
Return(Best_pop)
```

Fig. 5. GP-COACH pseudo-code.

2. Once the offspring population has been created it is joined to the current population, thus creating a new population whose size is double the current population size. Individuals in this new population are then sorted by their fitness, and the token competition diversity mechanism is applied. Irrelevant individuals (those not seizing any token) are deleted from the population, and secondary rules (new specific rules covering those examples whose tokens have not been seized during token competition) are created in order to increase the diversity of the population, if they are necessary.
3. The global fitness score is then calculated for this new population. If this score outperforms the one for the best population, the best population must be replaced by this new population (and the best global fitness score must also be updated). In any case, the new population becomes the current population in order to be able to apply the evolutionary process again.

The evolutionary process ends when the stop condition is verified (the maximum number of fitness evaluations is reached). Then the population kept as the best one is returned as the solution to the problem.

A pseudo-code summarizing GP-COACH algorithm is shown in Fig. 5.

4. Experimental framework

The performance of GP-COACH is analyzed using 24 classification data sets obtained from the UCI Repository of machine learning databases[2] [6] and the DELVE project.[3] The most important characteristics of these classification data sets are summarized in Table 3.

We have compared GP-COACH results with the ones obtained by different methods which use either an a priori feature selection process or an embedded one, in order to learn FRBCSs with a good interpretability-accuracy trade-off:

- **PCA-Ravi**: Ravi et al. [58] develop a process for deriving fuzzy rules for high-dimensional classification problems. This approach uses a reduced set of features extracted from the original ones by principal component analysis [72], and a modified threshold accepting algorithm [59] to build a compact rule subset.
- **2SLAVE**: González et al. propose in [38] 2SLAVE, a GA-based method for the learning of DNF fuzzy rules which follows the IRL approach to encoding rules in the population and includes an embedded feature selection process.
- **GP-PITT-Tsakonas**: Tsakonas [68] designs a GP-based FRBCS learning process which uses a context-free grammar to generate complete rule sets for each individual in the population.
- **GCCL-Ishibuchi**: Ishibuchi et al. propose in [45] a method which follows the GCCL approach to encoding rules in the population. This method uses a fixed-size rule population and "*Don't Care*" symbols for generalizing fuzzy rules. The rule consequents are an output class and a certainty factor. These latter are derived using a heuristic procedure prior to fitness evaluation, and the GA operates on the rule antecedent only.

[2] We have used two different versions of the HillValley data set, corresponding to the data without (1) and with (2) noise, respectively.

[3] URL: http://www.cs.toronto.edu/~delve/.

Table 3
Data sets characteristics.

Name	N. Inst.	N. Feat.	N. Clas.	Name	N. Inst.	N. Feat.	N. Clas.
Bupa	345	6	2	Cleveland	297	13	5
Ecoli	336	7	8	Flare	1066	9	2
Glass	214	9	6	HillValley1	1212	100	2
HillValley2	1212	100	2	Iris	150	4	3
Libras Mov.	360	90	15	Magic	19020	10	2
Page-blocks	5472	10	5	Parkinsons	195	22	2
Pen-based	10992	16	10	Pima	768	8	2
Quadruped	5000	46	4	Ringnorm	7400	20	2
Satimage	6435	36	6	Segment	2310	19	7
Sonar	208	60	2	Spambase	4597	57	2
Twonorm	7400	20	2	Wdbc	569	30	2
Wine	178	13	3	Yeast	1484	8	10

- **FRBCS_GP**: A GP-based method for learning FRBCSs proposed earlier by the authors [8]. The main differences between FRBCS_GP and GP-COACH are the following:
  - FRBCS_GP does not use a two level hierarchical inference process, and therefore it learns rule sets containing only one type of fuzzy rule.
  - The size of the current population does not change during the evolutionary process. GP-COACH uses a non-constant population size.
  - FRBCS_GP does not use any type of fitness score to determine the best-evolved population.
  - It uses a crisp fitness function, which uses the number of positive and negative examples, and not a fuzzy one as GP-COACH does.
  - Finally, FRBCS_GP uses a ranking selection scheme to select parents from the population, and different genetic operators to generate new children.

The parameters of the algorithms used are presented in Table 4. All the parameters for the algorithms are the ones recommended by the respective authors. In GP-COACH, the weights for the global fitness score have been heuristically determined, trying to represent a compromise between accuracy, simplicity of the individual rules and simplicity of the whole RB. It must be noted that those weights used were the same for all the data sets considered in the experimentation.

To develop the different experiments we consider a *10-fold cross-validation* model, and due to all the methods in study are non-deterministic ones, we have used three different seeds for each different partition. Therefore, for each data set we consider the average results for 30 runs. However, the results obtained from this validation are not completely independent, so the results neither present normal distribution nor homogeneity of variance. In this situation we consider the use of non-parametric tests, according to the recommendations made by Demšar in [31].

As such, these non-parametric tests can be applied to classification accuracies, error ratios or any other measure for technique evaluation, even including model sizes. Empirical results suggest that non-parametric tests are also more powerful

Table 4
Parameters used in the algorithms.

Algorithm	Parameters
PCA-Ravi	$PCA\_threshold = 75\%,\ U = 0.95\%,\ thresh = 0.035,\ thrtol = 10^{-8}$ $eps = 0.35,\ acc = 10^{-6},\ old = 9999,\ itrmax = 100,\ W_{NCP} = 10,$ $W_S = 1$
2SLAVE	$itrmax = 1000,\ itr\_no\_improve = 50,\ Pop = 20,\ \lambda = 0.8,\ k_1 = 0,$ $k_2 = 1, P_{AND} = 0.1,\ P_{OR} = 0.1,\ P_c = 0.6,\ P_m(per\ gene) = 0.05,$ $P_{Rotation} = 0.05$
GP-PITT-Tsakonas	$max\_individual\_size = 650nodes,\ Pop = 2000,\ itrmax = 10000$ $Tournament = 6,\ P_c = 0.35,\ P_m(per\ node) = 0.4,\ P_{Shrink} = 0.6$
GCCL-Ishibuchi	$Eval = 20{,}000,\ Pop = 100,\ N_{rep} = 20,\ P_c = 1.0,\ P_m = 0.1,$ $P_{don\prime tcare} = 0.9$
FRBCS_GP	$Eval = 20{,}000,\ Pop = 200,\ P_c = 0.5,\ P_m = 0.4,\ P_{dp} = 0.1,$ $min\_support = 0.01$
GP-COACH	$Eval = 20{,}000,\ Pop = 200,\ \alpha = 0.7,\ P_c = 0.5,\ P_m = 0.2,$ $P_{dp} = 0.15,\ P_i = 0.15, Tournament = 2,\ w_1 = 0.8,$ $w_2 = w_3 = 0.05$ and $w_4 = 0.1$

than the parametric ones. Demšar recommends a set of simple, safe and robust non-parametric tests for statistical comparisons of classifiers. In particular, we have considered two alternative methods based on non-parametric tests for analyzing the experimental results:

- Application of Friedman's test, Iman–Davenport's test and Holm's method as post hoc procedure. The two first tests may be used to see whether there are significant statistical differences among the algorithms in a certain group. If differences are detected, then Holm's test is employed to compare the best algorithm (control algorithm) against the remaining ones.
- Utilization of Wilcoxon's matched-pairs signed-ranks test. With this test, the results of two algorithms may be compared directly.

A wide description on the use of the non-parametric tests can be found in [31,34].

5. Experimental analysis of GP-COACH

In this section, an experimental study is carried out in order to demonstrate the suitability of some GP-COACH components:

- the token competition diversity mechanism,
- the insertion and dropping condition genetic operators, and
- the use of a GP-based algorithm instead of a GA-based one.

First of all, due to two different FRMs, the classical (*Max*) and the normalized sum (*Sum*), have been used in all our experiments, it is necessary to compare GP-COACH using both FRMs in order to find the best configuration.

We have applied a Wilcoxon signed-ranks test to find the best FRM (the statistical analysis is carried out considering the accuracy in test). In Table 5, R^+ and R^- are the rankings for the classical and the normalized sum FRMs, respectively. The best configuration (highlighted in bold) will be that one with the highest value.

Wilcoxon's test detects significant differences in the use of normalized sum FRM. Therefore, we will only consider this FRM for analyzing the different GP-COACH components.

We have analyzed the following GP-COACH components:

1. *Influence of the token competition diversity mechanism*: A comparative study between GP-COACH algorithm and a modified one not containing this diversity mechanism (called *GP-COACH No Token Comp.*) has been carried out.
2. *Effectiveness of insertion and dropping condition genetic operators*: A modified GP-COACH algorithm not containing these genetic operators (called *GP-COACH No Inser. No Drop.*) has been developed.
3. *Advantages of using a GP-based algorithm for learning fuzzy rules instead of a traditional GA*: A new GA-based algorithm for the learning of DNF fuzzy rules (called *GA-COACH*), has been implemented. This GA algorithm codifies a single DNF fuzzy rule per chromosome using a binary codification. A uniform crossover operator and a mutation operator changing the value of a gene randomly have been used as genetic operators (insertion and dropping condition specific operators have been not considered).

Accuracy results are shown in Table 7 ($\overline{\%Tra}$ and $\overline{\%Test}$ are the accuracy percentage for training and test data, respectively).

In Table 6, we show the statistical analysis carried out using the Wilcoxon signed-ranks test (considering the accuracy in test). R^+ is the ranking for the traditional GP-COACH algorithm, while R^- are the rankings for the different modified approaches. The best configuration (highlighted in bold) will be that one with the highest value. As it can be seen, Table 6 shows the robustness of our traditional GP-COACH algorithm, due to it is statistically better than all these modified approaches.

6. Comparative analysis with others FRBCS learning methods

In this section, the accuracy and compactness of GP-COACH is statistically analyzed by comparing its results with the ones obtained by other well-known FRBCS learning proposals.

First of all, we have to find the best FRM configuration for each one of the FRBCS learning methods considered in the comparative (with the exception of the Tsakonas method, where its own FRM has been used [68]). To do this we have applied a

Table 5
Wilcoxon's test for the best FRM configuration in GP-COACH, $p = 0.05$.

R^+ (FRM Max)	R^- (FRM Sum)	Critical value	Sig. dif.?
7.5	**292.5**	81	Yes

Table 6
Wilcoxon's test for analyzing some GP-COACH components, $p = 0.05$.

GP-COACH$_{Sum}$ vs.	R^+	R^-	Critical value	Sig. dif.?
GP-COACH$_{Sum}$No Token Comp.	**300.0**	0.0	81	Yes
GP-COACH$_{Sum}$No Inser.No Drop.	**265.5**	34.5	81	Yes
GA-COACH$_{Sum}$	**262.0**	38.0	81	Yes

Table 7
GP-COACH components analysis results.

Dataset	GP-COACH$_{Max}$		GP-COACH$_{Sum}$		GP-COACH$_{Sum}$		GP-COACH$_{Sum}$		GA-COACH$_{Sum}$	
					No Token Comp.		No Inser.	No Drop.$_{Sum}$		
	$\overline{\%Tra}$	$\overline{\%Test}$	$\overline{\%Tra}$	$\overline{\%Test}$	$\overline{\%Tra}$	$\overline{\%Test}$	$\overline{\%Tra}$	$\overline{\%Test}$	$\overline{\%Tra}$	$\overline{\%Test}$
Bupa	66.94	61.58	69.04	63.63	58.54	57.13	65.40	60.64	66.12	62.10
Cleveland	63.91	54.31	64.93	55.23	53.84	53.52	63.14	56.79	67.60	52.75
Ecoli	83.10	77.52	83.58	77.72	50.64	50.31	77.80	75.01	69.19	65.88
Flare	67.70	67.38	67.70	67.45	59.02	58.81	65.14	64.41	66.73	66.01
Glass	69.68	65.63	71.26	65.33	49.21	47.93	61.44	56.88	60.87	57.05
HillValley1	53.76	52.56	53.96	52.89	51.67	50.80	53.29	51.82	55.30	53.05
HillValley2	55.59	52.99	55.68	53.99	50.63	49.56	55.26	54.15	56.78	54.76
Iris	97.78	97.56	97.78	97.56	93.90	90.67	97.14	97.56	94.89	94.00
Libras Mov.	73.88	45.28	74.23	45.56	17.69	15.93	74.85	50.00	93.02	57.87
Magic	78.82	78.78	79.78	79.82	65.64	65.64	76.45	76.32	75.75	75.74
Page-blocks	90.39	90.30	91.30	91.23	89.78	89.78	91.25	91.19	90.51	90.47
Parkinsons	88.60	84.62	89.74	86.48	75.48	75.57	87.88	84.97	86.38	82.01
Pen-based	78.78	78.77	82.29	82.20	35.40	35.44	78.84	78.54	74.56	74.18
Pima	75.93	73.90	77.02	74.37	65.01	65.06	76.93	73.68	74.60	73.33
Quadruped	100.00	100.00	100.00	100.00	77.26	77.25	100.00	100.00	99.28	98.67
Ringnorm	86.95	86.75	91.24	91.13	64.22	64.10	77.68	77.40	86.67	86.10
Satimage	66.58	66.51	72.83	72.50	41.06	40.96	72.40	72.25	64.27	64.02
Segment	84.87	84.78	86.55	85.96	42.42	42.66	74.32	73.62	66.41	66.32
Sonar	76.16	67.12	80.25	67.46	62.27	58.42	78.49	65.83	79.72	60.40
Spambase	77.34	76.60	83.17	82.80	63.56	63.64	76.06	76.10	88.44	81.07
Twonorm	77.25	76.70	85.42	84.83	78.61	78.17	85.16	84.77	88.03	88.25
Wdbc	94.17	92.02	95.09	93.90	88.68	87.81	93.12	91.97	92.57	90.51
Wine	97.94	94.31	98.96	95.10	89.57	83.89	96.36	89.66	91.84	89.48
Yeast	41.84	40.67	49.74	48.56	34.33	33.65	42.46	41.71	43.77	42.34
Average	77.00	73.61	79.23	75.65	60.77	59.86	75.87	72.72	76.39	71.93

Table 8
Wilcoxon's test for the best FRM configurations, $p = 0.05$.

Algorithm	R^+ (FRM Max)	R^- (FRM Sum)	Critical value ($p = 0.05$)	Sig. dif.?
PCA-Ravi	149.0	**151.0**	81	No
2SLAVE	86.5	**213.5**	81	No
GCCL-Ishibuchi	**197.0**	103.0	81	No
FRBCS_GP	21.5	**278.5**	81	Yes

Wilcoxon signed-ranks test (the statistical analysis is carried out considering the accuracy in test). In Table 8, R^+ and R^- are the rankings for the classical and the normalized sum FRMs, respectively. The best configurations (highlighted in bold) will be those that present the highest values.

Wilcoxon's test only detects significant differences, in the use of one FRM or another, with regard to FRBCS_GP method. However, for our empirical study we will consider for each method the FRM configuration that presents the highest value.

Our study has been divided into three different parts: The first one comprises a statistical comparison of the accuracy of the six FRBCS learning methods considered in the empirical study. The statistical analysis of the compactness and interpretability results is carried out in the second part. Finally, a third part summarizes the GP-COACH accuracy and compactness performance.

6.1. Accuracy analysis

Accuracy results are shown in Table 9. Statistical analysis is carried out considering the accuracy in test.

Table 9
GP-COACH and other FRBCS learning methods accuracy results.

Dataset	PCA-Ravi$_{Sum}$		2SLAVE$_{Sum}$		GP-PITT-Tsakonas		GCCL-Ishibuchi$_{Max}$		FRBCS_GP$_{Sum}$		GP-COACH$_{Sum}$	
	$\overline{\%Tra}$	$\overline{\%Test}$	$\overline{\%Tra}$	$\overline{\%Test}$	$\overline{\%Tra}$	$\overline{\%Test}$	$\overline{\%Tra}$	$\overline{\%Test}$	$\overline{\%Tra}$	$\overline{\%Test}$	$\overline{\%Tra}$	$\overline{\%Test}$
Bupa	67.08	54.46	64.75	58.58	59.80	56.45	60.36	58.27	64.49	62.20	69.04	63.63
Cleveland	79.97	49.24	54.18	46.19	61.93	56.46	62.83	54.15	60.91	56.69	64.93	55.23
Ecoli	86.19	55.46	58.18	57.49	45.77	43.94	74.47	71.17	81.22	76.75	83.58	77.72
Flare	67.65	66.48	43.54	42.64	67.90	67.23	67.31	65.92	64.81	64.36	67.70	67.45
Glass	74.18	46.56	49.29	44.39	48.03	45.12	69.63	60.69	61.28	56.61	71.26	65.33
HillValley1	52.35	51.48	52.52	51.76	50.86	49.97	20.26	20.02	50.43	49.78	53.96	52.89
HillValley2	52.00	50.85	52.53	51.21	50.57	49.20	28.62	28.00	51.28	50.69	55.68	53.99
Iris	94.27	88.44	94.67	94.67	54.10	48.44	95.55	94.67	97.65	97.11	97.78	97.56
Libras Mov.	74.83	42.41	33.15	25.83	10.08	5.28	28.70	20.74	56.24	47.69	74.22	45.56
Magic	77.47	77.59	74.23	74.29	64.89	64.79	76.02	76.02	74.59	74.51	79.78	79.82
Page-blocks	91.15	90.70	91.40	91.42	93.03	92.92	90.41	90.34	91.24	91.09	91.30	91.23
Parkinsons	88.20	73.63	84.10	81.75	77.87	74.53	84.22	83.27	86.86	85.75	89.74	86.48
Pen-based	81.36	81.81	81.32	81.16	44.86	44.67	82.53	82.18	75.74	75.53	82.29	82.20
Pima	81.77	68.31	67.00	66.45	65.85	64.28	70.37	69.11	74.79	73.16	77.02	74.37
Quadruped	100.00	100.00	99.99	99.99	30.03	28.51	99.99	99.99	99.94	99.89	100.00	100.00
Ringnorm	38.40	30.12	80.12	79.64	50.87	50.51	91.81	91.70	94.10	93.84	91.24	91.13
Satimage	78.36	76.54	33.39	33.45	23.82	23.82	63.14	63.12	68.08	68.06	72.83	72.50
Segment	80.17	78.50	73.37	72.81	21.85	21.62	84.64	84.07	81.23	80.38	86.55	85.96
Sonar	92.90	27.65	78.45	70.72	65.24	52.42	83.49	72.40	83.30	71.15	80.25	67.48
Spambase	81.70	64.93	69.87	70.14	82.30	81.89	69.77	69.87	75.03	74.55	83.17	82.80
Twonorm	24.81	20.04	84.67	84.35	49.02	48.80	90.70	90.12	92.40	91.97	85.42	84.83
Wdbc	94.73	86.77	92.42	91.80	65.66	63.09	92.69	91.09	95.60	95.02	95.09	93.90
Wine	99.33	93.17	92.22	91.53	46.30	38.19	97.98	91.21	95.84	91.13	98.96	95.10
Yeast	58.94	39.45	15.22	14.51	32.36	31.76	50.69	49.01	52.79	52.16	49.74	48.56
Average	75.74	63.11	67.52	65.70	52.62	50.16	72.34	69.88	76.24	74.17	79.23	75.66

In Fig. 6 the values of the average rankings using Friedman's method are specified. Each column represents the average ranking obtained by an algorithm; that is, if a certain algorithm achieves rankings 1, 3, 1, 4 and 2 on five data sets, the average ranking is $\frac{1+3+1+4+2}{5} = \frac{11}{5}$. The height of each column is proportional to the ranking and *the lower a column is, the better its associated algorithm*. We then apply Friedman's and Iman–Davenport's tests (considering a level of significance $\alpha = 0.05$) to check whether differences exist among all the methods, presenting the results in Table 10.

Table 10 indicates that both Friedman's and Iman–Davenport's statistics are higher than their associated critical value, so the hypothesis of equivalence of results is rejected. A post hoc test is then needed in order to distinguish whether the control algorithm (GP-COACH, which obtains the lowest value of ranking computed through Friedman's test) is significantly better than the remainder. Table 11 shows all the possible hypotheses of comparison between the control algorithm and the others, ordered by their p-value and associated with their level of significance α.

Holm's method rejects all hypotheses. Therefore, according to Holm's procedure, the control algorithm (GP-COACH) is statistically better regarding accuracy than the rest of methods, with a p-value of 0.05.

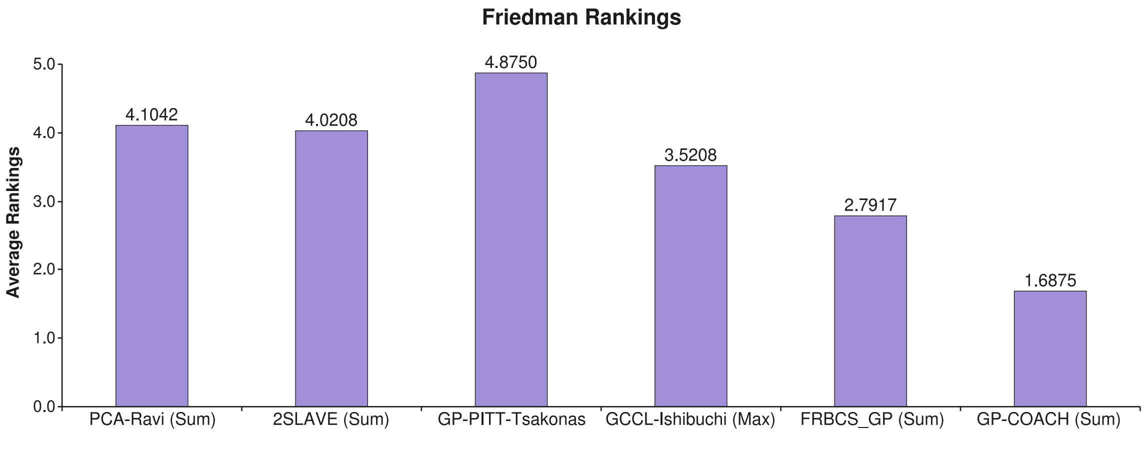

Fig. 6. Accuracy Friedman rankings.

Table 10
Statistics and critical values for Friedman's and Iman–Davenport's tests, $\alpha = 0.05$.

	Critical value	Hypothesis
Friedman's statistic		
43.2976	11.0705	Rejected
Iman–Davenport's statistic		
12.9832	2.2932	Rejected

6.2. Compactness analysis

Once we have demonstrated the good performance in term of accuracy of our proposal, in this second part we will analyze the compactness and interpretability results.

It must be pointed out that some of the FRBCS learning methods considered in the study learn DNF fuzzy rules, while other ones learn non-DNF (canonical) rules. As has been seen, a DNF fuzzy rule is a special type of rule which can comprise several canonical rules. Therefore it seems rather inappropriate to compare methods obtaining these two different types of rules. Due to the fact that GP-COACH learns DNF rules, we only will consider those methods which also learn DNF rules in this compactness study.

Compactness results are shown in Table 12, where $\overline{Rul}$ is the average rule number, $\overline{Var}$ is the average number of antecedent variables per rule, $\overline{Cond}$ is the average number of antecedent conditions per rule, and *CI* is a compactness index which has been calculated in order to measure the compactness and interpretability of a learned RB, with the following expression:

$$CI = Rul_N + Var_N + Cond_N, \tag{8}$$

Table 11
Holm table (GP-COACH is the control algorithm).

i	Algorithm	z	p	α/i	Hypothesis
5	GP-PITT-Tsakonas	5.9021	$3.5890 \cdot 10^{-9}$	0.0167	Rejected
4	PCA-Ravi$_{Sum}$	4.4748	$7.6484 \cdot 10^{-6}$	0.01	Rejected
3	2SLAVE$_{Sum}$	4.3205	$1.5568 \cdot 10^{-5}$	0.0125	Rejected
2	GCCL-Ishibuchi$_{Max}$	3.3947	$6.8710 \cdot 10^{-4}$	0.025	Rejected
1	FRBCS_GP$_{Sum}$	2.0445	0.0409	0.05	Rejected

Table 12
GP-COACH and other FRBCS learning methods compactness results.

Dataset	2SLAVE$_{Sum}$				FRBCS_GP$_{Sum}$				GP-COACH$_{Sum}$			
	$\overline{Rul}$	$\overline{Var}$	$\overline{Cond}$	*CI*	$\overline{Rul}$	$\overline{Var}$	$\overline{Cond}$	*CI*	$\overline{Rul}$	$\overline{Var}$	$\overline{Cond}$	*CI*
Bupa	4.57	4.96	14.55	1.40	18.53	4.83	12.40	1.31	10.07	1.38	2.49	0.16
Cleveland	12.37	9.19	21.00	1.10	34.53	3.82	10.12	0.52	23.83	3.05	7.44	0.35
Ecoli	10.37	4.49	11.35	0.97	30.40	3.31	7.44	0.70	25.57	2.88	6.21	0.57
Flare	2.97	5.98	12.39	0.94	15.23	3.47	9.42	0.55	8.13	1.80	4.11	0.19
Glass	8.80	6.32	14.69	1.07	23.47	3.71	8.16	0.63	17.43	2.64	5.56	0.39
HillValley1	6.63	21.38	48.74	0.32	26.93	16.10	50.54	0.30	7.27	3.18	7.65	0.04
HillValley2	6.40	47.73	157.42	0.88	34.33	8.22	24.39	0.16	6.90	3.20	8.16	0.04
Iris	3.93	2.72	6.71	0.96	3.00	1.51	2.69	0.28	3.23	1.22	1.75	0.13
Libras Mov.	25.53	37.93	73.35	0.65	49.77	3.75	7.09	0.16	113.93	53.36	76.03	1.07
Magic	4.23	6.31	15.03	0.96	33.33	5.53	13.82	0.83	9.33	1.71	4.33	0.16
Page-blocks	7.53	6.45	15.95	0.99	39.87	3.95	8.92	0.53	14.97	1.61	3.51	0.13
Parkinsons	3.43	9.96	18.86	0.64	7.53	4.23	10.87	0.29	6.40	1.67	3.77	0.09
Pen-based	39.97	11.76	26.55	1.12	87.07	3.25	7.67	0.26	89.70	4.27	9.35	0.35
Pima	3.80	4.24	10.60	0.74	27.90	4.46	11.76	0.88	17.23	2.46	5.15	0.36
Quadruped	6.00	18.22	29.01	0.53	4.27	1.24	1.75	0.01	4.57	1.24	1.55	0.01
Ringnorm	4.60	12.54	35.27	1.02	39.60	13.19	18.38	0.87	17.50	5.45	9.90	0.35
Satimage	9.83	18.34	40.45	0.77	93.40	11.00	25.59	0.47	27.53	5.82	13.29	0.22
Segment	10.47	9.21	21.58	0.73	38.8	6.30	14.00	0.48	23.30	3.28	6.85	0.21
Sonar	9.33	15.50	28.40	0.40	20.97	8.58	25.22	0.32	14.03	2.78	6.35	0.12
Spambase	7.90	22.25	49.19	0.59	43.93	13.46	31.57	0.37	10.27	3.77	7.48	0.08
Twonorm	24.40	14.02	42.74	1.22	99.27	16.85	49.08	1.46	51.67	4.11	9.15	0.27
Wdbc	5.47	9.90	21.24	0.50	16.30	3.57	9.02	0.18	4.90	1.17	3.03	0.03
Wine	5.73	6.55	15.87	0.77	9.60	3.60	7.98	0.40	7.57	1.90	4.65	0.18
Yeast	13.27	4.69	11.50	0.86	64.03	3.29	6.98	0.56	32.20	2.99	6.44	0.47

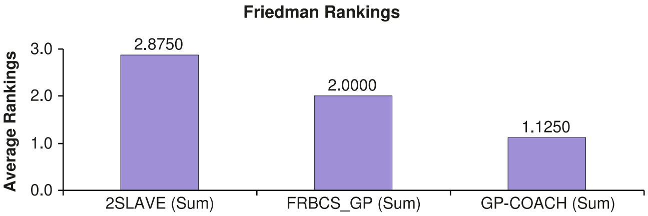

Fig. 7. Compactness Friedman rankings.

Table 13
Statistics and critical values for Friedman's and Iman–Davenport's tests, $\alpha = 0.05$.

	Critical value	Hypothesis
Friedman's statistic		
36.7500	5.9915	Rejected
Iman–Davenport's statistic		
75.1333	3.1996	Rejected

Table 14
Holm table (GP-COACH is the control algorithm).

i	Algorithm	z	p	α/i	Hypothesis
2	2SLAVE$_{Sum}$	6.0622	$1.3429 \cdot 10^{-9}$	0.025	Rejected
1	FRBCS_GP$_{Sum}$	3.0311	0.0024	0.05	Rejected

where Rul_N, Var_N and $Cond_N$ are the normalized values for three previous measurements. These normalized values have been obtained using the maximum and minimum values shown in Table 2.

The statistical analysis is carried out considering the CI measurement (the lower this is, the better the compactness).

In Fig. 7 the values of the average rankings using Friedman's method are specified. We then apply Friedman's and Iman–Davenport's tests (considering a level of significance $\alpha = 0.05$) to check whether differences exist among all the methods, presenting the results in Table 10.

Table 13 indicates that both Friedman's and Iman–Davenport's statistics are higher than their associated critical value, so the hypothesis of equivalence of results is rejected. A post hoc test is then needed in order to distinguish whether the control algorithm (the algorithm which obtains the lowest value of ranking computed through Friedman's test) is significantly better than the remainder. Table 14 shows all the possible hypotheses of comparison between the control algorithm and the others, ordered by their p-value and associated with their level of significance α.

Holm's method rejects all the hypotheses, which means that GP-COACH is statistically better regarding compactness than 2SLAVE and FRBCS_GP algorithms, with a p-value of 0.05.

6.3. Summary

Table 15 summarizes GP-COACH's performance with regard to the rest of the methods considered in the study. In this table, the symbols + , – or = show the existence or absence of significant differences between GP-COACH and the algorithm specified in the row via Holm's method, while the symbol * indicates that the existence or absence of significant differences is unknown because we have been unable to perform a statistical analysis.

Therefore, we can conclude that GP-COACH presents a good interpretability-accuracy trade-off, since it is the algorithm that presents the best ranking, and it statistically outperforms the rest of the methods in at least one of the two criteria (accuracy and compactness) analyzed.

Table 15
GP-COACH$_{Sum}$ performance.

Method	Acc.	Comp.	Method	Acc.	Comp.
PCA-Ravi$_{Sum}$	+	*	*2SLAVE$_{Sum}$*	+	+
GP-PITT-Tsakonas	+	*	*GCCL-Ishibuchi$_{Max}$*	+	*
FRBCS_GP$_{Sum}$	+	+			

Table 16
Fuzzy rule bases obtained for Iris dataset.

PCA-Ravi$_{Sum}$ (N. rules: 4, %Test: 73.33)
R^1 : If P_1 is $L_1^{1(2)}$ then *Class* is C^1
R^2 : If P_1 is $L_1^{2(2)}$ then *Class* is C^3
R^3 : If P_1 is $L_1^{4(5)}$ then *Class* is C^3
R^4 : If P_1 is $L_1^{2(3)}$ then *Class* is C^2
2SLAVE$_{Sum}$ (N. rules: 4, %Test: 93.33)
R^1 : If X_3 is $(L_3^1$ or $L_3^5)$ and X_4 is $(L_4^1$ or L_4^2 or $L_4^5)$ then *Class* is C^1
R^2 : If X_3 is $(L_3^1$ or L_3^2 or $L_3^5)$ and X_4 is $(L_4^1$ or $L_4^5)$ then *Class* is C^1
R^3 : If X_2 is $(L_2^1$ or L_2^2 or L_2^3 or $L_2^5)$ and X_3 is $(L_3^1$ or L_3^2 or L_3^3 or $L_3^5)$ and X_4 is $(L_4^3$ or $L_4^5)$ then *Class* is C^2
R^4 : If X_1 is L_1^4 and X_3 is $(L_3^4$ or $L_3^5)$ and X_4 is $(L_4^4$ or $L_4^5)$ then *Class* is C^3
GP-PITT-Tsakonas (N. rules: 49, %Test: 46.67)
R^1 : If X_1 is L_1^5 then *Class* is C^1
R^2 : If X_1 is L_1^4 then *Class* is C^3
...
R^{48} : If X_3 is L_3^3 then *Class* is C^2
R^{49} : If X_4 is L_4^2 then *Class* is C^1
GCCL-Ishibuchi$_{Max}$ (N. rules: 12, %Test: 86.67)
R^1 : If X_4 is L_4^1 then *Class* is C^1 with *CD* 1.0
R^2 : If X_4 is L_4^3 then *Class* is C^2 with *CD* 0.82
...
R^{11} : If X_2 is L_2^1 then *Class* is C^2 with *CD* 0.65
R^{12} : If X_1 is L_1^2 and X_3 is L_3^3 then *Class* is C^2 with *CD* 0.92
FRBCS_GP$_{Sum}$ (N. rules: 3, %Test: 93.33)
R^1 : If X_4 is L_4^1 then *Class* is C^1 with *CD* 1.0
R^2 : If X_2 is $(L_2^1$ or L_2^2 or L_2^3 or $L_2^5)$ and X_3 is L_3^3 and X_4 is L_4^3 then *Class* is C^2 with *CD* 0.95
R^3 : If X_4 is $(L_4^4$ or $L_4^5)$ then *Class* is C^3 with *CD* 0.85
GP-COACH$_{Sum}$ (N. rules: 3, %Test: 93.33)
R^1 : If X_4 is L_4^1 then *Class* is C^1 with *CD* 1.0
R^2 : If X_3 is $(L_3^2$ or $L_3^3)$ and X_4 is L_4^3 then *Class* is C^2 with *CD* 0.96
R_3 : If X_4 is $(L_4^4$ or $L_4^5)$ then *Class* is C^3 with *CD* 0.85

Finally, in Table 16, an example of the fuzzy rule bases obtained by the different methods for the Iris dataset is shown[4] (as an illustrative example of the mentioned trade-off).

7. Conclusions

In this paper we have presented GP-COACH, a genetic programming-based algorithm for the learning of COmpact and interpretable fuzzy rule bases, that also present a high ACcuracy on test data, for High-dimensional (a high number of features) classification problems. GP-COACH's main features are:

- It uses a context-free grammar which allows the learning of DNF fuzzy rules and the absence of some input features.
- It follows the *GCCL* approach that encodes a single rule per individual, and thus the RB is formed by the whole population.
- It includes a mechanism to increase the diversity in the population, *token competition*.
- It makes use of a two-level hierarchical inference process, which allows us to improve the accuracy of a learned RB, avoiding misclassification errors because of the use of specific rules (*secondary rules*) covering a small number of examples.
- Regarding FRBCS_GP, GP-COACH uses a new fitness function, new genetic operators and a global fitness score, and learns variable size rule sets with two different types of rules, obtaining more compact rule sets which also present a good accuracy performance.

An experimental study involving several high-dimensional data sets and another five well-known FRBCS learning algorithms has been carried out, and non-parametric statistical methods have been used to compare and analyze the compactness and accuracy of the algorithms. The main conclusions are the following:

[4] In PCA-Ravi$_{Sum}$ method, the variable "P_1" is an extracted one (by the Principal Analysis Components method) from the original variables (X_i). On the other hand, the numbers in brackets in the labels L indicate the number of fuzzy sets per variable, due to Ravi method simultaneously uses four fuzzy set partitions for each attribute with 2, 3, 4 and 5 linguistic labels, respectively.

- GP-COACH outperforms the rest of the algorithms with regard to the accuracy results on test data. It has been demonstrated to be able to obtain FRBCSs with a high generalization capability for high-dimensional problems.
- GP-COACH is able to learn compact and interpretable FRBCSs for high-dimensional problems.

As future work, it would be of interest to investigate the use of instance selection techniques [12] to deal with problems having high dimensionality because of the number of instances. We also would like to examine the possibility of hybridization of GP-COACH with those approaches which use hierarchical fuzzy partitions [29] and fuzzy partition context adaptation [28], with the aim of adapting fuzzy partitions to high-dimensional problems. Finally, we would analyze the behavior of GP-COACH for classifying imbalanced data sets [20,32].

References

[1] V. Akbarzadeh, A. Sadeghian, M.V.D. Santos, Derivation of relational fuzzy classification rules using evolutionary computation, in: Proceedings of the 2008 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE08), Hong Kong, China, 2008, pp. 1689–1693.

[2] M. Akbarzadeh-T, K. Kumbla, E. Tunstel, M. Jamshidi, Soft computing for autonomous robotic systems, Computers and Electrical Engineering 26 (1) (2000) 5–32.

[3] E. Alba, C. Cotta, J. Troya, Evolutionary design of fuzzy logic controllers using strongly-typed GP, Mathware and Soft Computing 6 (1999) 109–124.

[4] R. Alcalá, J. Alcalá-Fdez, F. Herrera, J. Otero, Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation, International Journal of Approximate Reasoning 44 (1) (2007) 45–64.

[5] J. Alcalá-Fdez, F. Herrera, F. Márquez, A. Peregrín, Increasing fuzzy rules cooperation based on evolutionary adaptive inference systems, International Journal of Intelligent Systems 22 (9) (2007) 1035–1064.

[6] A. Asuncion, D. Newman, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2007, <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

[7] F. Berlanga, M. del Jesus, M. Gacto, F. Herrera, A genetic-programming-based approach for the learning of compact fuzzy rule-based classification systems, in: 8th International Conference on Artificial Intelligence and Soft Computing (ICAISC'06), LNCS, vol. 4029, Springer-Verlag, 2006, pp. 182–191.

[8] F. Berlanga, M. del Jesus, F. Herrera, Learning fuzzy rules using genetic programming: context-free grammar definition for high-dimensionality problems, in: Proceedings of the I Workshop on Genetic Fuzzy Systems (GFS'05), Granada, Spain, 2005, pp. 136–141.

[9] J. Bezdek, S. Pal (Eds.), Fuzzy Models for Pattern Recognition. Methods that Search for Structures in Data, IEEE Press, New York, 1992.

[10] A. Botta, B. Lazzerini, F. Marcelloni, D. Stefanescu, Context adaptation of fuzzy systems through a multi-objective evolutionary approach based on a novel interpretability index, Soft Computing 13 (5) (2009) 437–449.

[11] E.V. Broekhoven, V. Adriaenssens, B.D. Baets, Interpretability-preserving genetic optimization of linguistic terms in fuzzy models for fuzzy ordered classification: an ecological case study, International Journal of Approximate Reasoning 44 (1) (2007) 65–90.

[12] J. Cano, F. Herrera, M. Lozano, Evolutionary stratified training set selection for extracting classification rules with trade-off precision-interpretability, Data and Knowledge Engineering 60 (2007) 90–108.

[13] B. Carse, A. Pipe, Introduction: genetic fuzzy systems, International Journal of Intelligent Systems 22 (9) (2007) 905–907.

[14] J. Casillas, B. Carse, Special issue on genetic fuzzy systems: recent developments and future directions, Soft Computing 13 (5) (2009) 417–418.

[15] J. Casillas, B. Carse, L. Bull, Fuzzy-XCS: a michigan genetic fuzzy system, IEEE Transactions on Fuzzy Systems 15 (4) (2007) 536–550.

[16] J. Casillas, O. Cordón, M. del Jesus, F. Herrera, Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction, IEEE Transactions on Fuzzy Systems 13 (1) (2005) 13–29.

[17] J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.), Accuracy Improvements in Linguistic Fuzzy Modeling, Studies in Fuzziness and Soft Computing, vol. 129, Springer-Verlag, 2003.

[18] J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.), Interpretability Issues in Fuzzy Modeling, Studies in Fuzziness and Soft Computing, vol. 128, Springer-Verlag, 2003.

[19] J. Casillas, F. Herrera, R. Pérez, M. del Jesus, P. Villar, Special issue on genetic fuzzy systems and the interpretability-accuracy trade-off, International Journal of Approximate Reasoning 44 (1) (2007) 1–3.

[20] M.-C. Chen, L.-S. Chen, C.-C. Hsu, W.-R. Zeng, An information granulation based data mining approach for classifying imbalanced data, Information Sciences 178 (16) (2008) 3214–3227.

[21] Z. Chi, H. Yan, T. Pham, Fuzzy Algorithms with Applications to Image Processing and Pattern Recognition, World Scientific, 1996.

[22] B.-C. Chien, J. Lin, T.-P. Hong, Learning discriminant functions with fuzzy attributes for classification using genetic programming, Expert Systems with Applications 23 (1) (2002) 31–37.

[23] W. Combs, J. Andrews, Combinatorial rule explosion eliminated by a fuzzy rule configuration, IEEE Transactions on Fuzzy Systems 6 (1) (1998) 1–11.

[24] O. Cordón, R. Alcalá, J. Alcalá-Fdez, I. Rojas, Genetic fuzzy systems: what's next? An introduction to the special section, IEEE Transactions on Fuzzy Systems 15 (4) (2007) 533–535.

[25] O. Cordón, M. del Jesus, F. Herrera, A proposal on reasoning methods in fuzzy rule-based classification systems, International Journal of Approximate Reasoning 20 (1999) 21–45.

[26] O. Cordón, M. del Jesus, F. Herrera, M. Lozano, MOGUL: a methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach, International Journal of Intelligent Systems 14 (11) (1999) 1123–1153.

[27] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific, 2001.

[28] O. Cordón, F. Herrera, P. Villar, Generating the knowledge base of a fuzzy rule-based system by the genetic learning of data base, IEEE Transactions on Fuzzy Systems 9 (4) (2001) 664–667.

[29] O. Cordón, F. Herrera, I. Zwir, Linguistic modeling by hierarchical systems of linguistic rules, IEEE Transactions on Fuzzy Systems 10 (1) (2002) 2–20.

[30] M. Delgado, E. Nagai, L.R. de Arruda, A neuro-coevolutionary genetic fuzzy system to build soft sensors, Soft Computing 13 (5) (2009) 481–495.

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

[32] A. Fernández, S. García, M. del Jesus, F. Herrera, A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets, Fuzzy Sets and Systems 159 (18) (2008) 2378–2398.

[33] M. Gacto, R. Alcalá, F. Herrera, Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems, Soft Computing 13 (5) (2009) 419–436.

[34] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization, Journal of Heuristics 15 (2009) 617–644.

[35] A. Geyer-Schulz, Fuzzy Rule-Based Expert Systems and Genetic Machine Learning, Physica-Verlag, Heidelberg, 1995.

[36] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.

[37] A. Gómez-Skarmeta, F. Jiménez, G. Sánchez, Improving interpretability in approximative fuzzy models via multiobjective evolutionary algorithms, International Journal of Intelligent Systems 22 (9) (2007) 943–969.

[38] A. González, R. Pérez, Selection of relevant features in a fuzzy genetic learning algorithm, IEEE Transactions on Systems, Man and Cybernetics, Part B 31 (3) (2001) 417–425.

[39] J. González, I. Rojas, H. Pomares, L. Herrera, A. Guillén, J. Palomares, F. Rojas, Improving the accuracy while preserving the interpretability of fuzzy function approximators by means of multi-objective evolutionary algorithms, International Journal of Approximate Reasoning 44 (1) (2007) 32–44.

[40] S. Gunal, R. Edizkan, Subspace based feature selection for pattern recognition, Information Sciences 178 (19) (2008) 3716–3726.

[41] F. Herrera, Genetic fuzzy systems: taxonomy, current research trends and prospects, Evolutionary Intelligence 1 (2008) 27–46.

[42] F. Hoffmann, D. Schauten, S. Hölemann, Incremental evolutionary design of TSK fuzzy controllers, IEEE Transactions on Fuzzy Systems 15 (4) (2007) 563–577.

[43] A. Homaifar, D. Battle, E. Tunstel, G. Dozier, Genetic programming design of fuzzy controllers for mobile robot path tracking, International Journal of Knowledge-Based Intelligent Engineering Systems 4 (1) (2000) 33–52.

[44] H. Ishibuchi, T. Nakashima, T. Morisawa, Voting in fuzzy rule-based systems for pattern classification problems, Fuzzy Sets and Systems 103 (2) (1999) 223–238.

[45] H. Ishibuchi, T. Nakashima, T. Murata, Design of accurate classifiers with a compact fuzzy-rule base using an evolutionary scatter partition of feature space, IEEE Transactions on Systems, Man and Cybernetics, Part B 29 (5) (1999) 601–618.

[46] H. Ishibuchi, T. Nakashima, M. Nii, Classification and Modeling with Linguistic Information Granules: Advance Approaches to Linguistic Data Mining, Springer-Verlag, 2004.

[47] H. Ishibuchi, Y. Nojima, Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning, International Journal of Approximate Reasoning 44 (1) (2007) 4–31.

[48] H. Ishibuchi, T. Yamamoto, Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining, Fuzzy Sets and Systems 141 (1) (2004) 59–88.

[49] H. Ishibuchi, T. Yamamoto, Rule weight specification in fuzzy rule-based classification systems, IEEE Transactions on Fuzzy Systems 13 (4) (2005) 428–435.

[50] R. Kohavi, G. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1–2) (1997) 273–324.

[51] P. Kouchakpour, A. Zaknich, T. Bräunl, Population variation in genetic programming, Information Sciences 177 (17) (2007) 3438–3452.

[52] P. Kouchakpour, A. Zaknich, T. Bräunl, Dynamic population variation in genetic programming, Information Sciences 179 (8) (2009) 1078–1091.

[53] J. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, The MIT Press, Cambridge, MA,USA, 1992.

[54] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering 17 (3) (2005) 491–502.

[55] R. Mendes, F.B. de Voznika, A. Freitas, J. Nievola, A genetic-programming-based approach for the learning of compact fuzzy rule-based classification systems, in: Principles of Data Mining and Knowledge Discovery: 5th European Conference (PKDD'01), LNCS, vol. 2168, Springer-Verlag, 2001, pp. 314–325.

[56] M. Mucientes, J.C. Vidal, A. Bugarín, M. Lama, Processing time estimations by variable structure TSK rules learned through genetic programming, Soft Computing 13 (5) (2009) 497–509.

[57] Y. Nojima, H. Ishibuchi, I. Kuwajima, Parallel distributed genetic fuzzy rule selection, Soft Computing 13 (5) (2009) 511–519.

[58] V. Ravi, P. Reddy, H. Zimmermann, Pattern classification with principal component analysis and fuzzy rule bases, European Journal of Operational Research 126 (3) (2000) 526–533.

[59] V. Ravi, H. Zimmermann, Fuzzy rule based classification with FeatureSelector and modified threshold accepting, European Journal of Operational Research 123 (1) (2000) 16–28.

[60] L. Sánchez, J. Corrales, Niching scheme for steady state GA-P and its application to fuzzy rule based classifiers induction, Mathware and Soft Computing 7 (2–3) (2000) 337–350.

[61] L. Sánchez, I. Couso, Advocating the use of imprecisely observed data in genetic fuzzy systems, IEEE Transactions on Fuzzy Systems 15 (4) (2007) 551–562.

[62] L. Sánchez, I. Couso, J. Corrales, Combining GP operators with SA search to evolve fuzzy rule based classifiers, Information Sciences 136 (1–4) (2001) 175–191.

[63] L. Sánchez, J. Otero, Boosting fuzzy rules in classification problems under single-winner inference, International Journal of Intelligent Systems 22 (9) (2007) 1021–1034.

[64] L. Sánchez, J. Otero, I. Couso, Obtaining linguistic fuzzy rule-based regression models from imprecise data with multiobjective genetic algorithms, Soft Computing 13 (5) (2009) 467–479.

[65] M. Setnes, R. Babuška, Rule base reduction: some comments on the use of orthogonal transforms, IEEE Transactions on Systems, Man and Cybernetics, Part B 31 (2001) 199–206.

[66] S. Smith, A Learning System Based on Genetic Algorithms, Ph.D. Thesis, University of Pittsburgh, 1980.

[67] P. Thrift, Fuzzy logic synthesis with genetic algorithms, in: Proceedings of the 4th International Conference on Genetic Algorithms (ICGA'91), San Diego, USA, 1991, pp. 509–513.

[68] A. Tsakonas, A comparison of classification accuracy of four genetic programming-evolved intelligent structures, Information Sciences 176 (6) (2006) 691–724.

[69] L. Wang, J. Mendel, Generating fuzzy rules by learning from examples, IEEE Transactions on Systems, Man, and Cybernetics 22 (6) (1992) 1414–1427.

[70] S. Wilson, Classifier fitness based on accuracy, Evolutionary Computation 3 (2) (1995) 149–175.

[71] M. Wong, K. Leung, Data Mining using Grammar based Genetic Programming and Applications, Kluwer Academics Publishers, 2000.

[72] M.-L. Zhang, J.M. Peña, V. Robles, Feature selection for multi-label naive Bayes classification, Information Sciences 179 (19) (2009) 3218–3229.