International Journal of Computational Intelligence Systems, Vol. 5, No. 2 (April, 2012), 254-275

Reducing the Complexity of Genetic Fuzzy Classifiers in Highly-Dimensional Classification Problems

Dimitris G. Stavrakoudis
Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki
Thessaloniki, 54124, Greece
E-mail: jstavrak@auth.gr

Georgia N. Galidaki
School of Forestry and Natural Environment, Aristotle University of Thessaloniki
Thessaloniki, 54124, Greece
E-mail: galidaki@for.auth.gr

Ioannis Z. Gitas
School of Forestry and Natural Environment, Aristotle University of Thessaloniki
Thessaloniki, 54124, Greece
E-mail: igitas@for.auth.gr

John B. Theocharis
Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki
Thessaloniki, 54124, Greece
E-mail: theochar@ eng.auth.gr

Received 19 November 2010
Accepted 1 June 2011

Abstract

This paper introduces the Fast Iterative Rule-based Linguistic Classifier (FaIRLiC), a Genetic Fuzzy Rule-Based Classification System (GFRBCS) which targets at reducing the structural complexity of the resulting rule base, as well as its learning algorithm's computational requirements, especially when dealing with high-dimensional feature spaces. The proposed methodology follows the principles of the iterative rule learning (IRL) approach, whereby a rule extraction algorithm (REA) is invoked in an iterative fashion, producing one fuzzy rule at a time. The REA is performed in two successive steps: the first one selects the relevant features of the currently extracted rule, whereas the second one decides the antecedent part of the fuzzy rule, using the previously selected subset of features. The performance of the classifier is finally optimized through a genetic tuning post-processing stage. Comparative results in a hyperspectral remote sensing classification as well as in 12 real-world classification datasets indicate the effectiveness of the proposed methodology in generating high-performing and compact fuzzy rule-based classifiers, even for very high-dimensional feature spaces.

Keywords: Genetic fuzzy rule-based classification systems (GFRBCS), local feature selection, genetic tuning, hyperspectral image classification, highly-dimensional classification problems.

1. Introduction

Over the past decade, the growing development of hyperspectral sensor technologies and the consequent commercial availability of hyperspectral satellite imagery have shifted the interest of the respective research community towards the evaluation of hyperspectral data for remote sensing classification tasks. Hyperspectral sensors collect several (typically 200 or more) narrow spectral bands, from the visible to the short-wave infrared portions of the electromagnetic

spectrum, providing an almost continuous spectral reflectance signature. Contrary to multispectral data, hyperspectral data proved capable of producing both genus-level and species-level classifications[1]. Particularly, in land cover classification of forests, where typically different species of the same genus coexist, it has been shown that hyperspectral satellite imagery can significantly increase the classification accuracy[2].

The high number of features involved in classification tasks from hyperspectral data necessitates the use of more sophisticated classifiers than the simple statistical ones, commonly employed in multispectral classification problems. In this direction, various methods from the field of pattern recognition and artificial intelligence have been proposed, including neural networks[3], fuzzy clustering algorithms[4], decision trees[5], kernel-based techniques[6], and combinations of them[7, 8]. In particular, support vector machines (SVMs) have been extensively used for high-dimensional datasets because of their good classification performance[9, 10], rendering them the reference classifier in most remote sensing task nowadays. Apart from the accuracy of the resulting thematic map, the remote sensing research community has recently focused on the interpretability of the considered classification model. Towards this direction, rule-based classifiers[11] and decision trees[12, 13] have been considered. Such models provide a better understating of the underlying physical relations of the classification problem at hand, which is useful from an operational remote sensing perspective, especially for hyperspectral imagery, where prior knowledge is rather limited.

Most of the aforementioned classifiers have indeed proved to be generally robust in dimensionality issues, but their application in very highly-dimensional feature spaces in prone to degrade their classification accuracy, a phenomenon commonly referred to as *Hughes' phenomenon*. Although recent research implies that such a behavior is also dependent on the training set size[14], the traditional approach is to apply a feature selection pre-processing step, so as to filter out the most irrelevant features[10, 15]. However, the application of a feature pre-filtering techniques is a non-trivial issue, as it entails a number of side-effects:

- Feature selection techniques measure the relevancy of each feature through some statistics metric. However, the selected features subset is not the optimal one for all classifiers, as each one creates the decision boundaries differently.
- Various land cover types are typically best discriminated though different features subsets. Hence, local feature selection techniques (that is, on a per class basis) are more well-suited for remote sensing classification tasks.
- Most feature selection methods include a threshold parameter which actually controls the selected number of features. The designer of the system is typically responsible for selecting this threshold value.

The last point is a crucial one, as it implies that the classifier under consideration must be trained for various feature subsets in order to determine the best possible one. This iterative procedure does, however, impose a considerable burden in the training process, from the computational requirements point view.

Fuzzy rule-based classification systems (FRBCSs) can provide a good balance between model simplicity, interpretability and classification accuracy. On the one hand, the fuzzy partition of the feature space can describe the naturally overlapping spectral signatures of closely related vegetation species more accurately, as compared to crisp (non-fuzzy) classifiers, resulting in high performing classifiers. On the other hand, FRBCSs can naturally encompass the local feature selection concept, by omitting any irrelevant feature variables from the antecedent part of the rules. However, traditional deterministic learning methods cannot appropriately handle highly-dimensional feature spaces, a fact that has limited so far the use of FRBCSs in multispectral data[16–18]. In the direction of efficiently determining the structure of FRBCSs for complex classification tasks, the enhanced search capabilities of Genetic Algorithms[19] (GAs) have been extensively used in the derivation of FRBCSs (and fuzzy rule-based systems (FRBSs) in general), giving rise to the field of genetic FRBCSs[20] (GFRBCSs). Feature selection mechanisms can easily be employed in GFRBSs, leading to compact fuzzy rule bases, thus increasing the inherent interpretability properties of FRBSs.

In a previous work[21], we proposed a GFRBCS for hyperspectral classification tasks, following the traditional approach of simultaneously selecting the relevant features and the fuzzy sets formulating the antecedent part of the fuzzy rules. This is generally a reasonable approach, since these two objectives are

closely related. However, as the dimensionality of the feature space increased, we observed that the simplicity of the final model and the computational requirements of its learning algorithm deteriorated, although the carefully crafted feature selection scheme employed alleviated the problem to a considerable degree. In this paper, we propose a methodology that decouples these two steps, that is, a local feature selection process (in a per rule basis) precedes the determination of the required fuzzy sets for the antecedent part of the fuzzy rule, which is performed in the previously selected subset of features. Hence, the complexity of the final model is reduced, along with the time requirements of its learning algorithm, as it will become apparent from our experimental study.

The rest of the paper is organized as follows. Section 2 summarizes the basic concepts that will be used throughout the paper. In Section 3, the various stages of the proposed system's learning algorithm are detailed. Experimental results using a hyperspectral satellite image are presented in Section 4, while Section 5 presents a thorough comparative analysis using 12 real-world classification datasets. The paper concludes in Section 5, with a summary of the proposed system and an outline of future research.

2. Basic Concepts

Assuming an N-dimensional feature space $\boldsymbol{x} = [x_1, \ldots, x_N] \in \Re^N$ and a set of M classes $\boldsymbol{C} = \{C_1, \ldots, C_M\}$, the proposed method constructs fuzzy rules of the form

$$\begin{aligned} R_k\text{: IF } x_1 \text{ is } A_1^k \text{ and} \cdots \text{ and } x_N \text{ is } A_N^k \\ \text{THEN } y \text{ is } C^k \text{ with } r^k, \end{aligned} \tag{1}$$

where A_i^k ($i = 1, \ldots, N$) are fuzzy sets defined along the ith input variable (feature) and r^k is the certainty degree of the classification in the class $C^k \in \boldsymbol{C}$, for a pattern belonging to the fuzzy subspace defined in the antecedent part of the rule.

For each input variable we assume here an associated term set of N_L possible values, represented by uniformly distributed triangular fuzzy sets with a linguistic meaning, resulting in a *descriptive*[20] FRBCS. Moreover, we use fuzzy rules of the so-called *disjunctive normal form* (DNF), where each variable is allowed to take as value multiple linguistic labels from its associated term set, joined by the OR disjunctive operator, instead of only one. As mentioned in the introduction, some input variables in each rule are allowed be absent, which, in fuzzy terms, has the meaning of a "don't care" fuzzy set (a fuzzy set with a unity membership grade for its entire universe of discourse). For the sake of simplicity, in the rest of the paper we will refer to an input variable (feature) x_i as *active* when the fuzzy clause "x_i is A_i^k" is present in the rule and as *inactive* in the opposite case.

Given a feature vector $\boldsymbol{x}^p$ to be classified, the matching degree of the kth rule is derived through

$$\mu^k(\boldsymbol{x}^p) = \underset{i=1}{\overset{N}{\wedge}} \left\{ \underset{q=1}{\overset{\ell_i^k}{\vee}} \mu_i^{L_i^q}(x_i^p) \right\}, \tag{2}$$

where $\mu_i^{L_i^q}(x_i^p)$ is the membership grade of each linguistic term participating in the formulation of ith variable's fuzzy value, and $\wedge$ and $\vee$ denote the AND and OR operators, respectively. The AND operator is implemented conventionally through the minimum operator and the OR operator is implemented through the bounded sum, defined for two membership values a and b as

$$\text{bs}(a,b) = \min(1, a+b). \tag{3}$$

Hence, the composite fuzzy set is mathematically equivalent to a trapezoidal fuzzy set, having a far more reasonable interpretation than the one provided by the usually selected maximum operator, which results in taking the envelop of neighboring membership functions.

The fuzzy reasoning method is implemented here through

$$C_{\max} = \underset{j=1,\ldots,M}{\arg\max} \sum_{R^k | C^k = j} \mu^k(x^p) \cdot r^k \tag{4}$$

using the so-called maximum voting scheme[22]. A thorough discussion of various voting schemes can be found in Ref. 23, whereas other fuzzy reasoning methods can be found in Refs. 24 and 25. Finally, assuming a set of labeled patterns $S = \{(\boldsymbol{x}^p, c^p),\ p = 1, \ldots, Q\}$, we calculate the confidence values r^k in this paper through

$$r^k = \frac{\sum_{p | c^p = C^k} \mu^k(\boldsymbol{x}^p)}{\sum_{p=1}^{Q} \mu^k(\boldsymbol{x}^p)}. \tag{5}$$

2.1. *Genetic fuzzy rule-based systems*

Since the middle of the nineties, GFRBCSs (and GFRBSs in general) have found considerable application in many and diverse application areas[26]. GFRBCSs combine the high interpretability properties of FRBCSs with the enhanced search capabilities of Evolutionary Algorithms (EAs), in order to automatically extract an optimal fuzzy rule base. EAs are a genre of universal optimization methods inspired from the genetic adaptation of natural evolution, among which GAs [19] are the most celebrated one. They can effectively attack multiple objectives simultaneously and are well-known for their ability to avoid local minima. These attributes render GFRBCSs attractive solutions in hyperspectral classification tasks, as they can produce high-performing classifiers, consolidating feature selective properties at the same time.

In the previous years, various genetic learning approaches have been considered for creating GFRBCSs[22, 27–31], each exhibiting different benefits and drawbacks. Here we concentrate on the so-called *iterative rule learning* (IRL) approach[29–31], which is the methodology followed by the proposal of this paper. Under the IRL, a rule extraction algorithm (REA) in repeatedly invoked, iteratively adding fuzzy rules to the rule base, one at a time. Those training examples that are sufficiently covered by the current rule set are removed from the training set, so that subsequent invocations of the rule generation algorithm will concentrate on the remaining uncovered instances. In evaluating a single rule at a time, the IRL methodology considerably reduces the dimensionality of the search space, splitting the problem into smaller, easier to handle sub-problems. Therefore, the derivation of the whole rule base is easier and faster, especially when a large number of features and/or training patterns are considered. Prominent examples of GFRBSs following the IRL approach are the MOGUL[29] and the SLAVE[30–31] systems.

2.2. *Boosting fuzzy rule-based classifiers*

Under the IRL, each time a new rule is generated, through an invocation to the REA, well-covered examples are completely removed from the training set. Since rules generated at later stages are unaware of the previously removed training instances, they may conflict with previously derived rules, an effect termed *cooperation vs. competition* problem. To overcome this problem, the use of AdaBoost has been proposed for designing GFRBCSs under the IRL approach[32–33]. AdaBoost[34] is the most well-known boosting algorithm and its main idea is to combine a set of low quality classifiers with a voting scheme, in order to generate an overall classifier that performs better than any of its constituents alone. Accordingly, each fuzzy rule can be regarded as a simple but partial (in terms of overall classification performance) classifier. The algorithm assigns a weight to every example of the training set and each time a new rule is generated, the examples covered by the new rule are effectively down-weighted, according to their matching degree. Because instances are never completely removed from the training set, subsequent rules are indirectly aware of the previously derived ones, while at the same time, the algorithm concentrates on uncovered patterns, which have retained their initial, higher, weight values.

Assuming a training set S_{trn}, comprising Q elements $\boldsymbol{e}^p=(\boldsymbol{x}^p, c^p)$, $p=1,\ldots,Q$, the AdaBoost algorithm assigns a weight w^p to each one of the Q training patterns, initialized with a unity value $w^p = 1$. Each time a new rule R^k is produced, describing the class label $C^k \in \boldsymbol{C}$, the rule error $E(R^k)$ is computed, taking into consideration the patterns' current weights and their matching degrees:

$$E(R^k) = \frac{\sum_{p|c^p=C^k} w^p \cdot \mu^k(\boldsymbol{x}^p)}{\sum_{p=1}^{Q} w^p \cdot \mu^k(\boldsymbol{x}^p)}. \tag{6}$$

Incorrectly classified patterns retain their former weights, whereas correctly classified are reduced according to

$$w^p \leftarrow \begin{cases} w^p, & \text{if } C^k \neq c^p \\ w^p \cdot \beta^k(\boldsymbol{x}^p), & \text{if } C^k = c^p. \end{cases} \tag{7}$$

The factor

$$\beta^k(\boldsymbol{x}^p) = \left(\frac{E(R^k)}{1 - E(R^k)} \right)^{\mu^k(\boldsymbol{x}^p)} \tag{8}$$

depends on the current rule's error and each pattern's matching degree. Effectively, well-covered examples that are correctly classified are down-weighted according to their matching degree, whereas misclassified or uncovered patterns retain their former weights. Hence, subsequent invocations to the REA are biased in producing new rules in uncovered regions of

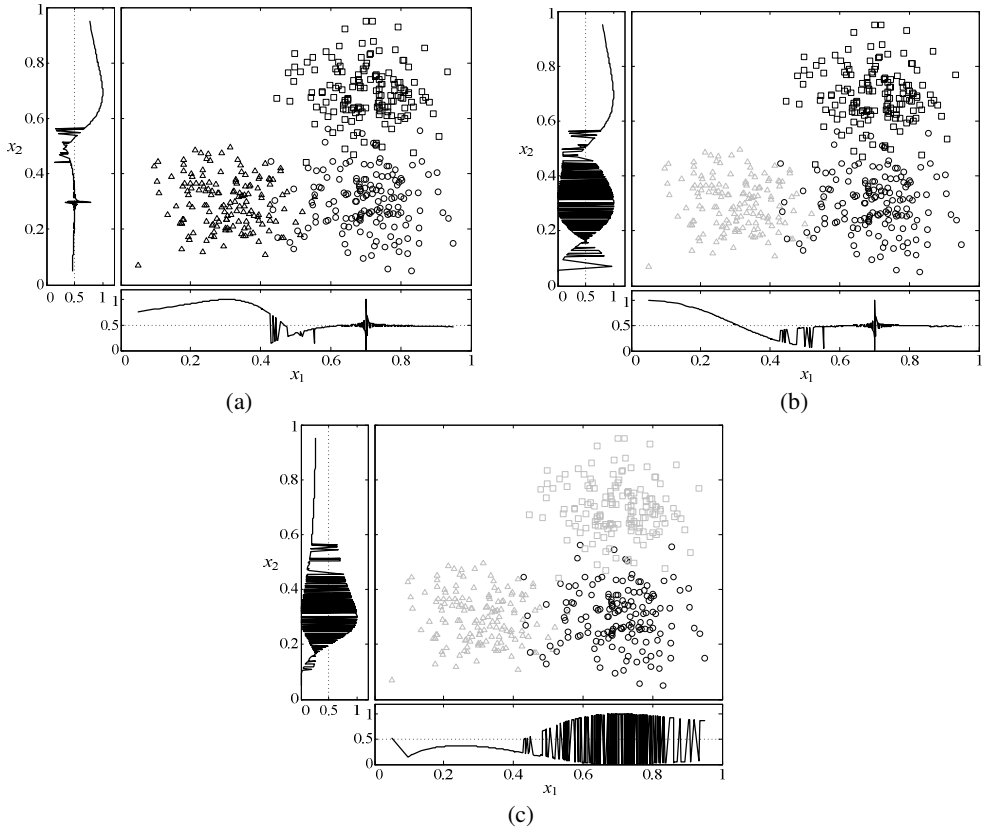

Fig. 1. FPV values in a synthetic classification dataset with two features and three classes: a) all patterns have unity weights, b) patterns of class △ have been down-weighted, and c) patterns of class □ have been down-weighted.

the features space, or in regions that are more difficult to learn (that is, mixed areas of the feature space).

Normally, the AdaBoost algorithm computes a confidence value for each partial classifier (based on its error), which is then used to infer a classification decision through a weighted voting scheme. However, these confidence degree can take any value, as contrasted to the confidence values r^k of the proposed FRBCS (5), which are confined in [0,1] and therefore provide a much clearer interpretation of each rule's importance. Hence, since the fuzzy reasoning method employed by our system (4) is actually a weighted voting scheme, we disregard AdaBoost's special aggregation scheme and exploit only its weighting scheme, as a means to induce the desired cooperation among the rules.

2.3. *Reinforcing the feature selection process*

The flexibility of the EAs allows an easy integration of the feature selection process in the REA, along with the selection of the fuzzy sets formulating the antecedent part of the rule. Previous proposals of GFRBCSs – designed under the IRL methodology – followed this approach[30–33, 35], through the inclusion of a binary string in the chromosomes' encoding scheme. However, as we have observed in our previous work[21], this approach fails when considering highly-dimensional feature spaces, as the ones encountered in hyperspectral remote sensing classification tasks. In order to improve the feature selection characteristics of the learning process, we proposed the inclusion of deterministic information in the REA, based on the notion of the so-called *feature partition vector*[36] (FPV). The FPV is a criterion that quantifies the degree to which each example of a set of Q labeled patterns can be correctly classified by a single feature, independently from all other features, and is defined – considering the ith feature – through

$$G(\boldsymbol{x}_i) = \left\{ g(x_i^1), \ldots, g(x_i^Q) \right\} \subset \Re^Q , \qquad (9)$$

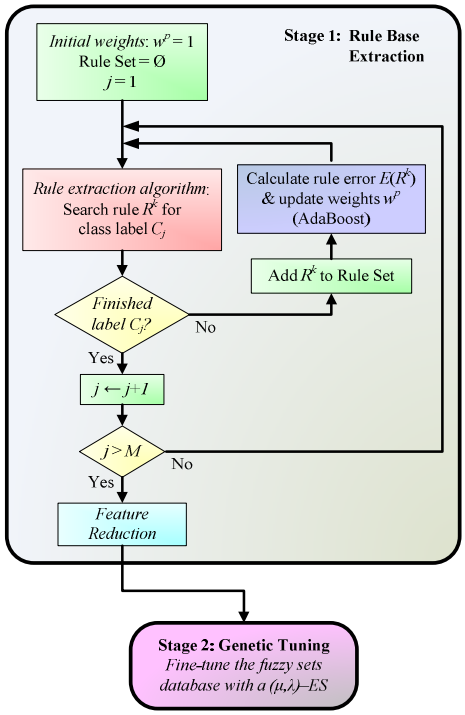

Fig. 2. Schematic representation of the proposed system's learning algorithm.

where $g(x_i^p)$ are membership degrees in [0,1], that can be derived from any classifier capable of producing fuzzy outputs. For computationally efficiency, we calculate these degrees here through the simple class allocation scheme used in fuzzy c-means (FCM):

$$g(x_i^p) = \left(\sum_{j=1}^{M} \frac{\left(x_i^p - v_i^{c^p}\right)^2}{\left(x_i^p - v_i^j\right)^2} \right)^{-1}, \qquad (10)$$

where v_i^j is the cluster center of the jth class label, along the ith feature variable. Hence, patterns which are located close to the cluster center of their class and far from other centers attain high membership grades, whereas patterns in mixed areas are assigned degrees close to zero. Particularly, an FPV value $g(x_i^p) < 0.5$ denotes that the pattern $\boldsymbol{e}^p$ will be misclassified if only the ith feature is considered. Since AdaBoost assigns weights to the training examples, the center of the jth class cluster is defined through the weighted average

$$v_i^j = \frac{\sum_{p|c^p=j} w^p \cdot x_i^p}{\sum_{p|c^p=j} w^p}. \qquad (11)$$

An example of the FPV membership grades is given in Fig. 1, considering a synthetic classification dataset with two features and three classes. Considering the initial unity weights (Fig. 1a), the two features x_1 and x_2 exhibit high memberships for values below 0.4 and above 0.6, respectively. This means that the patterns of the classes marked with ^ and □ signs can be identified by only considering features x_1 and x_2, respectively. Next, assume that the patterns of the class with a ^ marker are down-weighted, so that $w^{\hat{}} = 0.1$ (Fig. 1b). Now, the membership grades for the lower values of the second feature have increased. Since the patterns of the ^ class have not been completely removed, not all patterns of the ○ class attain high membership grades. Nonetheless, the sum of the respective membership grades is clearly higher than before. Finally, if the patterns of the □ are also down-weighted to $w^{\square} = 0.1$ (Fig. 1c), then only patterns belonging to the ○ class achieve high membership grades in both features.

Given a (possibly composite) fuzzy set A_i with a membership function $\mu_i(x_i)$, defined along the ith input variable, and a certain class label $C^k \in \boldsymbol{C}$, we define the *visibility degree* (VD) of the fuzzy set A_i as

$$\mathrm{VD}(A_i) = \frac{\sum_{p|c^p=C^k} w^p \cdot \mu_i(x_i^p) \cdot g(x_i^p)}{\sum_{p|c^p=C^k} w^p \cdot \mu_i(x_i^p)}. \qquad (12)$$

The VD expresses the degree to which the linguistic term A_i covers training patterns with high FPV values. Note that (12) is a typical fuzzy coverage criterion with respect to the FPV values. Although features with low VDs can also be useful (since the patterns could be discriminated in a multi-dimensional subspace), high VDs definitely denote that a single feature can correctly classify the equivalent portion of the training set, an attribute that will be evaluated throughout the proposed system's REA.

3. The Proposed Learning Algorithm

In this section we describe the proposed Fast Iterative Rule-based Linguistic Classifier (FaIRLiC), which is a GFRBCS designed under the principles of the boosted IRL methodology presented in Section 2.2. FaIRLiC is constructed through a two-stage learning algorithm, a schematic representation of which is depicted in Fig 2. During the rule base extraction (RBE) stage, an initial fuzzy rule base is produced, as described in Section 2.2. Similarly to the SLAVE algorithm[30, 31], we learn one

class label at a time. Each time a new rule is generated though an invocation to the REA, the weights of the patterns are re-weighted through (7) and the process is repeated until some termination conditions are fulfilled. We considered two such conditions:

(i) The new rule's fitness value is zero, which signifies the inability of the REA to find a new useful rule for the class label currently handled.

(ii) The new rule does not increase the classification performance of the rule base obtained so far significantly, as expressed by a pre-specified lower threshold t_r, selected as t_r = 1 % in this paper. The performance of the system is measured in a separate validation set of examples, which requires our initial set of labeled patterns to be split into three disjoint sets: the training, the validation, and the testing ones. Since the training patterns cannot be assumed to be equally distributed for all classes (and usually are not), this threshold value t_r is compensated on a per class basis through

$$t_r^{(j)} = M \cdot \frac{Q_j}{Q}, \quad j = 1,\ldots,M \ , \tag{13}$$

where Q_j is the number of training patterns belonging to the *j*th class label.

3.1. *Rule extraction algorithm*

As mentioned in the introduction, the main contribution of this paper lies in decoupling the feature selection process and linguistic terms selection ones. To succeed this goal, the REA is performed through four successive steps: i) prototype fuzzy cell localization, ii) feature selection, iii) linguistic terms selection and iv) linguistic terms reduction, which are detailed in the following.

3.1.1. Prototype fuzzy cell localization

Selecting the relevant features for each rule and the linguistic terms participating in its antecedent part are obviously two interrelated objectives. Therefore, prior to applying the feature selection process, we need to select some prototype linguistic terms in each input dimension. In order to select the best linguistic term for the *i*th input, we calculate for each linguistic term A_i^ℓ, $\ell = 1,\ldots,N_L$, the degree

$$D(A_i^\ell) = \sum\nolimits_{p|c^p=C^k} w^p \cdot \mu_i^{A_i^\ell}(x_i^p) \cdot g(x_i^p) \ , \tag{14}$$

taking into consideration the weights of the training patterns, their matching degree with the fuzzy set A_i^ℓ and their FPV values (9). The single linguistic term that maximizes (14) is selected for the *i*th input variable. Effectively, the initial hypercube defined by the antecedent part of the rule is located in the most interesting region of the feature space, as it covers with high matching degrees patterns that exhibit high FPV values and have high associated weights (that is, uncovered or previously misclassified patterns).

3.1.2. Feature selection

The second stage of the REA decides the active features of the rule through a typical binary GA[19], considering the previously selected linguistic terms in each dimension. The chromosomes comprise *N* bits, each representing if the respective variable will participate in the antecedent part of the rule. We use a binary tournament selection with elitism, two-point crossover, applied with a probability p_c, and uniform mutation with probability p_m per gene. The algorithm terminates on a maximum number of generations or if no better solution is found for a pre-specified number of generations.

The population is initialized randomly, using the VDs of each feature (12) as the random number distribution. Exceptionally, if for some chromosome the best VD is higher that 0.8, we inactivate all other features, leaving only one variable active. To increase the performance of the feature selection process (and subsequently the convergence of the GA), we consider two additional FPV-related mutation operators, applied with a probability 0.2 per chromosome:

- *Feature elimination*: The feature with the worst VD is deactivated, provided that the chromosome encodes a fuzzy rule with at least two active features.
- *Delete all but one features*: If the best VD among the active features is higher than 0.8, all other input variables are deactivated.

Three optimization criteria are considered for the fitness function, trying to minimize the number of features, maximize the weighted fuzzy number of positive examples $n^+ = \sum_{p|c^p=C^k} w^p \cdot \mu^k(\boldsymbol{x}^p)$ and minimize the weighted fuzzy number of negative examples $n^- = \sum_{p|c^p\neq C^k} w^p \cdot \mu^k(\boldsymbol{x}^p)$. The first criterion is defined as the ratio of inactive features to the total number of features:

$$\mathrm{IR} = \frac{N_i}{N-1}, \tag{15}$$

where N_i is the number of inactive features (genes with a zero value in the chromosome). The second criterion measures the rule's *consistency*, which is the degree to which the rule covers many positive training instances and few negative ones:

$$\mathrm{Cons} = \begin{cases} 0, & \text{if } n^+ < n^- \\ (n^+ - n^-)/n^+, & \text{otherwise.} \end{cases} \tag{16}$$

The last criterion is the class coverage of the rule, trying to maximize the number of positive examples covered. In this case however, a possible problem might arise from the inclusion of AdaBoost's weights in the definition of positive and negative examples. Conventionally, the AdaBoost algorithm encourages new classifiers (rules) to handle previously misclassified or uncovered patterns. It does not, however, discourages new classifiers from also covering previously down-weighted examples. On the contrary, such examples can reduce the error (6), even to a small degree. Although this behavior is indifferent when boosting other classifiers, it is undesirable for FRBCSs because interpretability issues demand a pattern to be covered by only one rule with a high matching degree, at least to the extent that this is possible. Hence, we use a modified coverage criterion, initially introduced in Ref. 35, which penalizes previously well-covered patterns. Each time a new rule is produced, positive patterns exhibiting matching degrees higher than 0.5 are assigned a token t^p, receiving the class label of the rule $t^p = C^k$. The modified coverage is defined as:

$$\mathrm{Cov}^{\mathrm{mod}} = \max\left(0, \frac{\sum_{p|c^p=C^k} v^p \cdot \mu^k(\boldsymbol{x}^p)}{\sum_{p|c^p=C^k} |v^p|}\right), \tag{17}$$

where

$$v^p = \begin{cases} w^p - 1, & \text{if } t^p = C^k \\ w^p, & \text{otherwise.} \end{cases} \tag{18}$$

Since the weights w^p lie in [0,1], v^p penalize previously well-covered patterns with a negative weight.

The GA aims at maximizing the above three criteria, which are all normalized in [0,1], with the total fitness function being defined as their product:

$$f_1 = \mathrm{IR} \cdot \mathrm{Cons} \cdot \mathrm{Cov}^{\mathrm{mod}}. \tag{19}$$

3.1.3. Linguistic terms selection

After having selected the relevant features of the rule, the third step of the REA identifies the DNF-type linguistic terms that will be included in its antecedent part, for each active feature. This step is also realized through a GA, using an integer encoding, with each (possibly composite fuzzy set) defined by a pair of genes. The minimum value of the two genes represents the first linguistic term, whereas the maximum one denotes the last linguistic term selected, with all the terms in between being joined with the OR disjunctive fuzzy operator. The GA is realized using a binary tournament selection with elitism, two-point crossover and Thrift's mutation, applied with the same probabilities as before. The termination conditions are the same as the ones in the feature selection step. Finally, the fitness function is defined as the product of the consistency (16) and modified class coverage (17) criteria:

$$f_2 = \mathrm{Cons} \cdot \mathrm{Cov}^{\mathrm{mod}}. \tag{20}$$

3.1.4. Linguistic terms reduction

The previous step's fitness function measures the value of each candidate solution (chromosome), though the patterns the related rule covers. It is therefore possible for the fuzzy rule to extend in empty regions of the feature space, where no training patterns are located. This behavior does not affect the performance of the desired model, but it is undesirable from its interpretability point of view, since over-general rules provide misleading information on the mapping from the feature space to the class space. Hence, as a last step in the REA, we apply a simple deterministic linguistic terms reduction step, which corrects any such deficiencies. Specifically, for each active input variable we remove one fuzzy term at a time (if there are more than two) and accept this change if the classification performance in both the training and validation sets does not decrease. The inner fuzzy sets of a contiguous block of linguistic terms are never removed, in order to avoid the creation of intermittent composite fuzzy sets.

3.2. *Feature reduction*

One of the objectives of the REA is to minimize the number of active features for each new rule produced, through the fitness function of the feature selection step

(15). However, this procedure relies on the stochastic application of genetic operators, which might lead to a somewhat inferior result. Hence, as a last step in the RBE stage (see Fig. 2), a features reduction algorithm is applied, similar to the linguistic terms reduction one, described above. In this step, for each rule we inactivate one input variable at a time and observe the performance of the system in the training and validation sets. If these performances do not degrade and the rule's consequent does not change, we accept the chance. Moreover, since removing a feature results in a generalization of the rule, we must also assure that the overlapping between rules describing the same class label does not increase, a fact that would negate any merits achieved through the modified coverage function (17). Here we measure the similarity between two fuzzy rules k and ℓ, describing the same class label C_j, through the matching degrees of the patterns they cover, a measure first introduced in Ref. 35:

$$S_{k\to\ell} = 1 - \frac{1}{N_e} \cdot \sum_{p|c^p=C_j} \frac{\left|\mu^k(\boldsymbol{x}^p) - \mu^\ell(\boldsymbol{x}^p)\right|}{\mu^k(\boldsymbol{x}^p) + \mu^\ell(\boldsymbol{x}^p)}, \quad (21)$$

where N_e is the number of patterns belonging to class C_j, for which either $\mu^1(\boldsymbol{x}^p)$ or $\mu^2(\boldsymbol{x}^p)$ is not zero. If the deactivation of one feature in the kth rule results in an increase of the similarity between any other with the same class consequent, then this change is reverted.

3.3. *Genetic tuning*

The second stage of FaIRLiC's learning algorithm targets at increasing the classification performance of the system, by fine-tuning the fuzzy sets definitions in each input variable. Tuning is performed in the global fuzzy sets database level, thus preserving the linguistic nature of the fuzzy rule base. The genetic tuning is performed with another type of evolutionary algorithm, called Evolution Strategies[37] (ES). ES are well-suited for real-valued optimization tasks, since they encode additional strategy parameters in each chromosome, adapting the algorithm itself throughout the evolution, along with the object variables. A detailed description of ES can be found in the aforementioned reference. Here, we use a typical (μ,λ)–ES, with the proposed population sizes, that is, a (15,100)–ES. In order to maintain the interpretability of the descriptive fuzzy rule base, we preserve the strong fuzzy partition of each input variable by adjusting only the modal points of the inner fuzzy sets. It is obvious that only the parameters of those variables that are used by at least one rule need to be included in the chromosome. The fitness function is based solely on the classification performance P_c of the system, using both the training and the validation pattern sets:

$$f_{tun} = 0.5 \cdot \left(P_c^{(trn)} + P_c^{(val)}\right). \quad (22)$$

The algorithm terminates if a maximum numbers of generations is reached (100 in this paper) or if no better individual is found in a fixed number of generations, selected as 30 in our experiments.

4. Application in Hyperspectral Remote Sensing Classification

Remote sensing classification from hyperspectral satellite images is an arduous task, because of the large number of features involved and the strong overlapping between the different class signatures. In this section, we will present the application of FaIRLiC in hyperspectral remote sensing classification, highlighting the importance of FRBCSs in such problems. The study area is the island of Thasos, Greece's most northerly island. Its surface area is 399 km^2, and its perimeter is approximately 102 km. Elevation ranges from sea level to 1217 m. *Pinus brutia* is the dominant tree species at lower elevations (0 to 800 m), whereas *Pinus nigra* is found at higher altitudes[38]. On August 1, 2003, a Hyperion image (level 1 radiometric product) covering a part of the island from north to south was acquired. Hyperion is the first spaceborne hyperspectral instrument to acquire both visible near infrared (VNIR) (400–1000 nm) and shortwave infrared (900–2500 nm) spectra[39]. The image exhibits 30 m spatial resolution and comprises a total of 242 bands in the aforementioned range of the electromagnetic spectrum, out of which 198 are the useful ones, whereas the rest of them contain no data. The Hyperion image was geometrically corrected using an orthorectified QuickBird image of the area.

Field survey was conducted to identify land cover classes, resulting in 147 plots (of minimum 30×30 m in size) being located with Global Positioning System coordinates. After careful photo-interpretation, using both the Hyperion image and the aforementioned QuickBird one, our training set was augmented to 1000 points and labeled into 6 classes: *Pinus brutia*, *Pinus nigra*, deciduous trees, other vegetation, non-vegetated

Table 1. Parameters used for the GAs of the REA.

Parameter	Value
Maximum number of iterations	1000
Numbers of iterations allowed without change	100
Population size	20
Mutation probability p_m (per gene)	$1/N_{\text{genes}}$
Crossover probability p_c	0.8

Table 2. Comparison between FeSLiC and FaIRLiC.

	FeSLiC	FaIRLiC
P_{BT} (%)	80.93	81.30
P_{AT} (%)	84.98	84.85
R	8.57	6.23
F/R	3.29	2.11
GU	27.07	12.77
L/V	1.75	1.60
Gen_1	618.9	332.5
Gen_2	86.3	79.7
T_1 (s)	141.9	23.3
T_2 (s)	114.2	36.0

P_{BT} (%) = Testing performance before tuning; P_{AT} (%) = Testing performance after tuning; R = number of rules; F/R = average number of features per rule; GU = number of features globally used; L/V = number of fuzzy labels per variable; Gen_1 = average number of generations per rule for the RBE; Gen_2 = number of generations for genetic tuning; T_1 (s) = RBE's execution time in seconds; T_2 (s) = genetic tuning's execution time in seconds.

Table 3. Performance of the rule base simplification procedures for FeSLiC and FaIRLiC.

		R	F/R	GU	L/V
FeSLiC	Without simplification	8.83	6.95	54.83	2.64
	With simplification	8.57	3.29	27.07	1.75
FaIRLiC	Without simplification	6.23	2.88	17.10	2.44
	With simplification	6.23	2.11	12.77	1.60

R = number of rules; F/R = average number of features per rule; GU = number of features globally used; L/V = number of fuzzy labels per variable.

areas, and water. The last class was visually identified in the image and was included in the classification scheme for the sake of completeness in the resulting thematic map. Prior to classification, the bands of the image were enhanced using a special image enhancement technique described in Ref. 21.

4.1. *Results obtained*

In order to derive unbiased conclusions, we used a 5-fold partition of our dataset and in each case the average results are reported. Due to FaIRLiC's algorithmic requirements, each initial training set was further split into two equal portions, that is, the training and validation sets. Moreover, since GFRBCSs employ stochastic processes (that is, the EAs), we performed six independent runs on each partition, using different random seeds. In each case, the average results over these 30 runs are given. The same procedure was followed with all GFRBCSs presented in this section. FaIRLiC was coded in C++ and all experiments were conducted on an Intel Core 2 Quad Q9650 at 3.0 GHz, with 4 GB of RAM. Table 1 reports the parameters used for the GA of the REA (both the feature selection and linguistic terms selection steps). Following Ref. 40, the

Table 4. Comparison of various GFRBCS.

	P (%)	R	F/R	GU	L/V
SGERD	75.21	7.50	2.00	12.37	1.00
2SLAVE-2	80.18	15.77	7.44	87.30	4.99
FeSLiC	84.98	8.57	3.29	27.07	1.75
FaIRLiC	84.85	6.23	2.11	12.77	1.60

P (%) = Testing performance; R = number of rules; F/R = average number of features per rule; GU = number of features globally used; L/V = number of fuzzy labels per variable.

Table 5. Comparison of FaIRLiC with non-fuzzy classifiers.

	Performance (%)
SVM	85.90
***k*NN**	81.60
C4.5	84.20
FaIRLiC	84.85

mutation probability has been set to the inverse number of genes in the chromosome.

In Ref. 21, we introduced FeSLiC, a GFRBCS constructed through a three stage process: the first stage produced an initial rule base under the principles of the boosted IRL described in Section 2.2, the second stage performed a deterministic rule base simplification, similar to the reduction steps presented in Section 3.1.4 and 3.2 with an additional rule reduction step, whereas the last stage was identical to FaIRLiC's genetic tuning stage. The major difference of the two systems lies in the REA, whereat FeSLiC performed the feature selection and linguistic terms selection in one step, through a hybrid representation scheme for its GA. In order to highlight the advantages of the proposed FaIRLiC, we first compared it against FeSLiC, with the averaged results given in Table 2. The table includes the testing classification performance before and after tuning, the number of rules, the average number of features per rule and the globally used features (features used by at least one rule), as well as the number of generations and the runtime in seconds for both the RBE stage and the genetic tuning one. For the FaIRLiC system, the number of generations per rule in the RBE stage is the sum of generations required for the feature selection and linguistic terms selection steps.

From the comparison of the two systems it becomes apparent that FaIRLiC results in far more compact rule bases, comprising 27% less rules, 36% less features per rule and less than half globally used features. Additionally, the time requirements for the RBE stage are remarkably reduced, with this stage being executed more than six times faster than FeSLiC's one. The reduced complexity of the initial fuzzy rule base also results in reduced computational requirements for the genetic tuning stage, which is performed more than three times faster. Most importantly, all these advantages of FaIRLiC are achieved without any penalty in the classification accuracy. Table 3 additionally compares the performance of the simplification stage for the two systems. Contrarily to FeSLiC, where this simplification is performed in an autonomous stage, FaIRLiC's rule base simplification procedures are performed inside the RBE loop, at least the linguistic terms reduction step. Nevertheless, we have measured the equivalent statistics as if those procedures had not been performed. From the comparison it becomes apparent that FeSLiC earns much more substantial gains from this stage. We can therefore conclude that the high number of features comprised in our task hinder the ability of FeSLiC's RBE stage to produce an optimal rule base. As a subsequent side-effect, the latter algorithm dissipates precious computational effort in balancing the feature and terms selection objectives, as it has become apparent from its runtime requirements.

To validate FaIRLiC's performance, we compared it against two other GFRBCSs of the literature, namely, the 2SLAVE-2[31] and the SGERD[41], tested using the Keel software package[42]. SGERD is a GFRBCS targeting at minimizing the structural complexity the classifier, as well as the computational requirements of

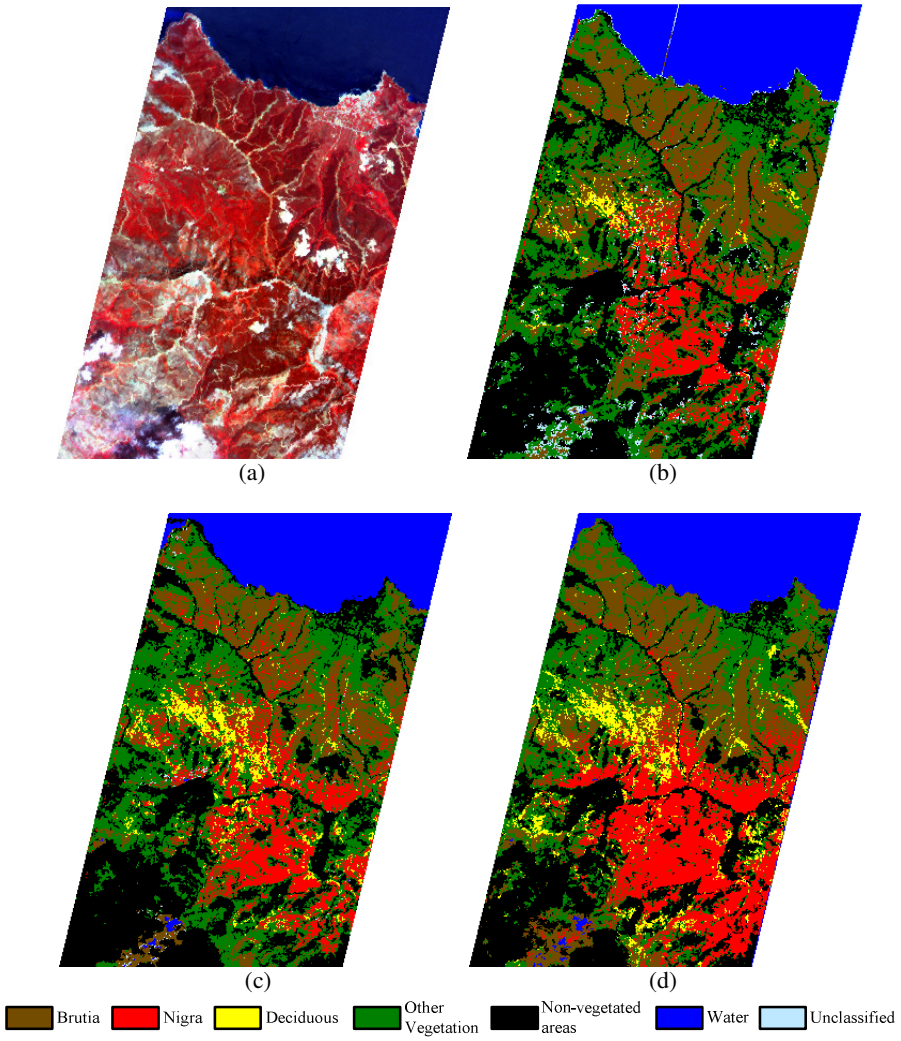

Fig. 3. A part of Thasos land cover: (a) 3 band composite, (b) FaIRLiC map, (c) FeSLiC map and (c) SVM map.

its learning algorithm. The obtained results are presented in Table 4. We can observe that the SGERD algorithm created fuzzy rule bases with a more or less similar structural complexity compared to FaIRLiC's one, at the cost, however, of the classification accuracy. On the other hand, the 2SLAVE-2 produced significantly larger structures, comprising two and a half times more rules and using almost seven times more features. To that extent, we can conclude that FaIRLiC results in high performing rule-based classifiers, maintaining a reasonably simple structure at the same time.

Table 5 compares FaIRLiC's performance with that of other non-fuzzy classifiers, commonly applied in remote sensing classification tasks from hyperspectral images. In particular, we have considered the SVM classifier, implemented with a Gaussian kernel, the *k*NN classifier, and Quinlan's C4.5[43], which constructs a classification decision tree. For the *k*NN algorithm, we tested k values in the range [1,40] and the value that maximized the algorithm's average performance on the validation set was finally chosen. The same technique was used to determine the parameters C and γ of the SVM classifier, considering a grid of possible values. The SVM classifier achieved the highest classification performance, although FaIRLiC and C4.5 also achieved comparable performance. Considering the computational efficiency of our model, as well as its

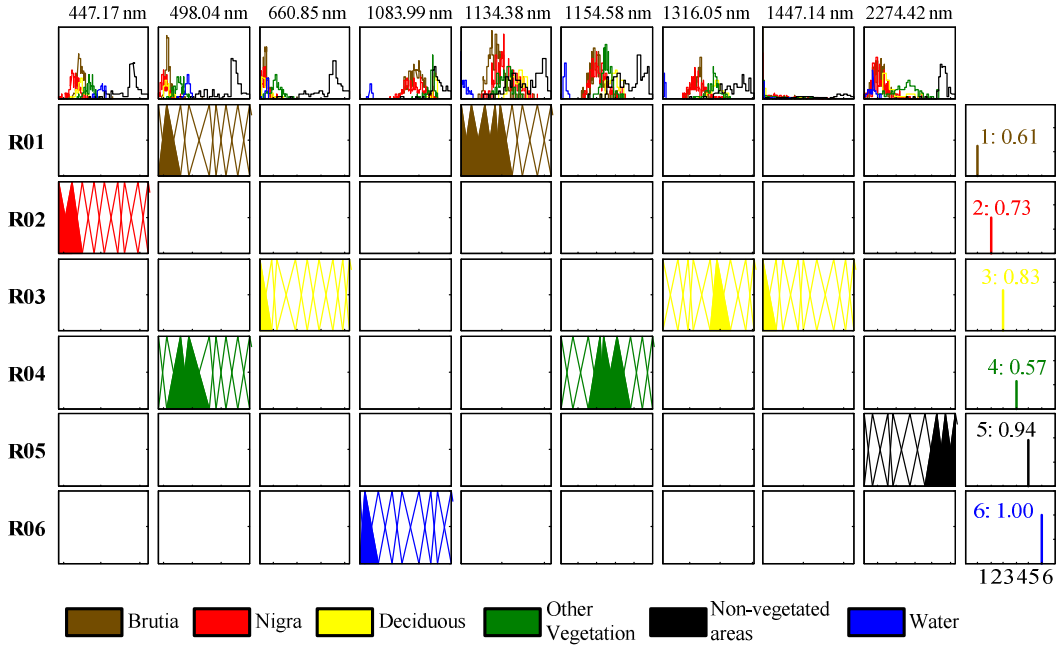

Fig. 4. A visual representation of the best FaIRLiC model after tuning.

linguistic interpretation, we can conclude that the proposed learning algorithm can achieve satisfactory classification performance. We must note that tuning is necessary (see Table 2 for FaIRLiC's performance before tuning), if we are to achieve comparable classification performance with other advanced classifiers.

Fig. 3 depicts the land cover maps obtained by the best (in terms of testing classification accuracy) FaIRLiC (Fig. 3b), FeSLiC (Fig. 3c) and SVM (Fig. 3d) classifiers. Fig. 3a also shows a three-band composite of the region, using bands with central wavelengths 752.43 nm, 660.85 nm and 477.69 nm as the three positioned components of RGB, respectively. The classification accuracy obtained by the models was 89.5%, 91% and 90.5%, respectively. Although FeSLiC's accuracy on the testing dataset was higher, we can observe that compared to FaIRLiC, the former classifier clearly overestimated the deciduous class in the center of the region. Moreover, more pixels in the north part of the island were misclassified as *Pinus nigra*, inside the *Pinus brutia* forest, compared to FaIRLiC. Similar conclusions can be derived for the SVM classifier, which again overestimated the deciduous class, as well as the *Pinus nigra* class, particularly in the south-east part of the region. From the presented comparison of the thematic maps we can conclude that FaIRLiC resulted in the most general result, a fact that can be attributed to the simple structure of the model (it comprises 6 rules and 9 globally active features).

4.2. *Advantages of FRBCSs*

The analysis presented so far has showed that FaIRLiC can produce simple fuzzy rule based classifiers, attaining equivalent performance to other non-fuzzy classifiers. However, one of the most important characteristics of FRBCSs is considered to be their interpretability. In order to highlight FRBCSs comparative advantages when applied in remote sensing classification tasks, we focus here on the best (in terms of classification accuracy) FaIRLiC model obtained, the thematic map of which was given in Fig. 3b. A visual representation of the obtained rule base after tuning is depicted in Fig. 4, where the first row of sub-plots also shows the class histograms (along with each band's central wavelength as header) and the last column represents the consequent of each rule, along with its confidence degree (5). Although this model obtained the highest classification accuracy in the testing dataset (89.5%), the resulting rule base is a very simple one, comprising only six rules (one for each class label), 1.67 features per rule on average and 9 globally active

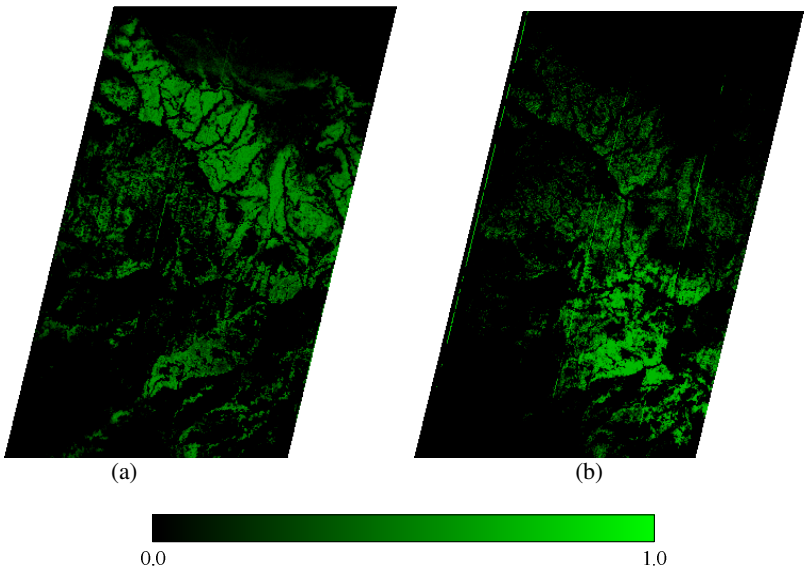

Fig. 5. Fuzzy images obtained by FaIRLiC for a) *Pinus brutia*, b) *Pinus nigra*.

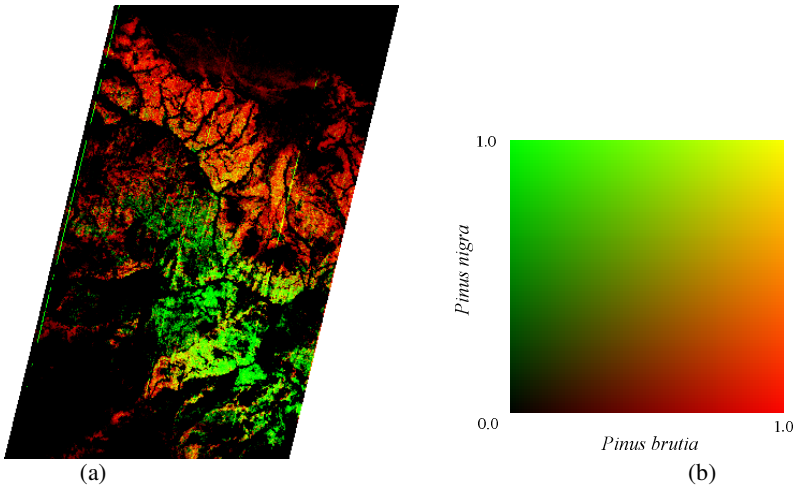

Fig. 6. a) Composite fuzzy images obtained by FaIRLiC for *Pinus brutia* and *Pinus nigra* and b) the respective color map.

features, with no rule having more than three active features.

One advantage of FRBCSs is that each rule can be evaluated independently from others, providing a degree in [0,1] which is the product of the matching degree and the rule's certainty degree $\mu^k(\boldsymbol{x}^p)\cdot r^k$. Applying this procedure in the whole image, we can derive fuzzy images for each class label, as the ones shown in Fig. 5a and 5b for the *Pinus brutia* and *Pinus nigra* classes, respectively. These images provide the confidence in the classification of each pixel, for the class under consideration, and are very useful from an operational point of view, when the objective is to identify the existence of a single class in the area under study. Comparing the fuzzy images of Fig. 5 with the thematic map obtained (Fig. 3b), it can be observed that through the fuzzy images the user can accurately locate the regions with high confidence values for each class label.

The antecedent part of each fuzzy rule can be view as a transformation $\Re^{N_a} \rightarrow [0,1]$, where N_a is the

Table 6. Datasets characteristics and the number of fuzzy sets per input variable used for FaIRLiC.

Dataset	Features	Classes	Patterns	Fuzzy Sets
korWet	53	5	1219	7
korAgro	53	8	2706	13
korWhole	53	13	3925	13
thasos	198	6	1000	9
wdcBands	191	7	420	9
wdcBandsSF	281	7	420	9
colon	2000	2	62	7
dlbcl	4026	2	45	5
lymphoma	4026	9	96	9
leukemia	7129	2	72	9
prostate	12600	2	102	5
lung	12600	5	203	7

number of the rule's active features. When seen from this scope, the rules of a FRBCS can provide useful tools for analyzing the mixing between different classes in the area under study. For example, the most difficult problem in our area of study is the discrimination between *Pinus brutia* and *Pinus nigra*, since these classes represent different species of the same genus. If we dispose the matching degrees (2) of the two rules representing these classes (first and second rule, respectively) in different channels of a color image, we can derive a composite fuzzy image, as the one given in Fig. 6a. The respective color map is shown in Fig. 6b: *Pinus brutia* pixels are represented by different levels of red color (according to their matching degree), whereas *Pinus nigra* is identified by different shades of green. Moreover, the mixed areas of the region (which normally exist in most physical ecosystems) are assigned yellow-like colors, as a result of the fusion between the two base colors. Note that in Fig. 6a, the matching degrees of each rule have been normalized in [0,1] for a clearer visualization, similarly to the histogram stretching of color images applied in remote sensing applications.

5. Comparative Analysis

The previous section presented the application of FaIRLiC in a hyperspectral remote sensing classification task, highlighting the comparative advantages of FRBCSs in equivalent problems. Nevertheless, FaIRLiC is a general purpose classifier, particularly aiming at handling high-dimensional feature spaces. To this end, we conducted additional comparative experimentation with 12 real-life datasets, the most important characteristics of which are summarized in Table 6. These datasets have been categorized into three groups, depending on the number of features they involve. The first category includes three (relatively) low-dimensional classification problems, referring to a remote sensing classification task from a multispectral IKONOS satellite image, in an area surrounding Lake Koronia in northern Greece. In addition to the original four bands of the image, advanced higher order spectral and spatial features were derived, resulting in a total of 53 features. In order to conduct additional studies from the quality of the ecosystem point of view, the area was carefully segmented into two zones: the wetland (korWet) and the agricultural (korAgro) ones. For the sake of completeness, the whole dataset is also considered (korWhole) here, comprising the union of the two zones. A detailed description of these datasets can be found in Ref. 44.

The second dataset category includes three remote sensing classification tasks with a medium number of features. Apart from the Thasos dataset presented in the previous section, we consider the hyperspectral data taken by the Airborne Hyperspectral Digital Imagery Collection Experiment (HYDICE) sensor[45]. The original image covers the region of the Washington DC Mall and has 210 bands covering the 0.4–2.4 μm spectral range. The water-absorption bands were discarded resulting in 191 bands, where 420 locations were selected and labeled into seven classes (wdcBands). Because this is an urban classification task, structural features (SF) can effectively improve the classification performance, as it has been proven by previous

Table 7. Training and testing classification accuracies obtained by the various classifiers considered. In each case, the maximum testing performance is highlighted in boldface.

Dataset	C4.5Rules		Ripper		SGERD		2SLAVE-2		FeSLiC		FaIRLiC	
	%Trn	%Tst	%Trn	%Tst	%Trn	%Tst	%Trn	%Tst	%Trn	%Tst	%Trn	%Tst
korWet	93.78	87.13	97.58	**88.11**	73.71	74.66	83.71	79.47	89.91	87.46	86.09	83.81
korAgro	74.90	66.70	91.41	68.55	61.21	61.05	62.01	66.78	75.59	**72.01**	70.68	68.64
korWhole	70.94	61.99	91.80	69.04	47.18	46.57	53.40	57.94	72.97	**69.22**	67.20	64.78
thasos	92.00	83.40	94.93	82.10	79.74	75.21	89.07	80.18	87.75	84.63	86.39	**84.85**
wdcBands	98.57	94.29	98.51	91.90	79.88	79.33	96.42	93.02	97.06	**95.79**	95.30	93.97
wdcBandsSF	99.46	95.24	99.29	91.19	86.24	84.40	99.67	95.83	98.54	96.51	97.75	**96.75**
colon	99.18	77.18	89.75	68.97	89.28	83.08	95.90	66.97	83.09	73.91	86.85	**83.14**
dlbcl	97.22	64.56	96.66	71.78	95.91	81.89	94.84	65.69	83.76	69.68	93.37	**85.50**
lymphoma	100.00	**91.99**	93.45	65.09	95.56	87.39	91.37	71.36	80.10	71.42	85.72	74.73
leukemia	98.24	77.33	97.54	80.29	–	–	66.43	52.67	84.01	74.21	94.75	**91.98**
prostate	97.79	78.38	98.04	84.38	–	–	68.35	51.63	77.32	71.81	89.87	**86.33**
lung	99.13	**92.01**	98.65	88.21	–	–	88.45	73.34	78.42	74.42	88.15	83.29
average	*93.44*	*80.85*	*95.63*	*79.13*	–	–	*82.47*	*71.24*	*84.04*	*78.42*	*86.84*	***83.15***

%Trn = Training performance (%); %Tst = Testing performance (%).

Table 8. Time requirements in seconds for the tested algorithms.

Dataset	C4.5Rules	Ripper	SGERD	2SLAVE-2	FeSLiC	FaIRLiC
korWet	3.00	7.60	1.17	213.33	263.47	48.44
korAgro	25.00	88.20	2.23	852.00	791.15	165.46
korWhole	29.20	219.20	3.57	2854.00	2058.43	369.81
thasos	5.60	11.60	5.93	532.23	256.09	59.36
wdcBands	1.20	4.80	3.07	171.20	95.91	28.47
wdcBandsSF	1.60	5.60	5.60	209.83	112.51	33.54
colon	6.40	6.20	91.57	67.73	46.97	9.50
dlbcl	17.40	15.00	379.90	129.03	72.85	10.76
lymphoma	19.20	17.40	482.20	760.63	505.99	115.29
leukemia	47.00	41.20	–	365.23	166.81	24.81
prostate	130.80	123.60	–	1060.50	355.07	101.10
lung	214.60	146.60	–	1628.27	1016.57	266.65
average	*41.75*	*57.25*	–	*737.00*	*478.48*	*102.77*

research[46]. Hence, we calculated the six statistical SF proposed in Ref. 46 for each one the 15 most important initial bands of the image, as derived by the SVM-FuzCoC feature selection algorithm[47]. The new dataset (wdcBandsSF) comprises a total of 281 features (191 bands + 6×15 SF).

The last dataset category comprises six high-dimensional microarray classification problems, related to cancer prediction through measurements retrieved from gene sequences. The first four (colon, dlbcl, lymphoma and leukemia) are available at http://www.upo.es/eps/aguilar/datasets.html, whereas the last two (lung and prostate) were downloaded from http://datam.i2r.a-star.edu.sg/datasets/krbd/index.html. Note that for the datasets in this category, the classifier must be able to identify the few truly informative features, since the vast majority of features are irrelevant and introduce noise to the classification task.

In this section, we will confine the comparative experimentation to interpretable classifiers and, in particular, to rule-based classifiers that have an equivalent description with FaIRLiC. To that extent, apart from the GFRCSs presented in the previous section, we will consider the C4.5Rules[43] and the Ripper[48] algorithms, which produce crisp rule bases in the form of IF–Then rules, where the antecedent part of the rules contains crisp descriptions instead of fuzzy sets. We have used a 5-fold cross validation procedure for all datasets, which was repeated six times in the case of the GFRBCSs. Table 6 includes the number of fuzzy labels per input variable used for FaIRLiC, which was determined as the one that maximizes its performance, through a trial-and-error procedure. The same number

Table 9. Compactness results for the various classifiers considered. For each dataset, the minimum CI is highlighted in boldface.

Dataset	C4.5Rules				Ripper				SGERD		
	R	F/R	GU	CI	R	F/R	GU	CI	R	F/R	GU
korWet	21.00	4.10	23.40	1.28	26.60	2.98	35.40	1.71	6.20	1.93	7.40
korAgro	40.80	5.33	39.60	1.33	111.80	3.52	52.20	2.10	11.60	1.95	14.20
korWhole	65.40	6.18	48.80	1.54	162.80	3.70	52.80	2.00	19.60	1.96	18.00
thasos	17.80	3.61	25.60	1.03	26.20	2.46	40.60	1.44	7.50	2.00	12.37
wdcBands	9.80	2.32	7.60	0.25	14.40	1.87	18.20	1.25	7.97	2.00	13.33
wdcBandsSF	9.00	2.79	7.00	**0.45**	14.80	1.78	19.00	1.39	8.20	1.99	13.97
colon	4.80	1.47	2.80	1.03	4.60	1.25	4.20	0.93	2.40	2.00	4.20
dlbcl	3.40	1.13	1.40	0.64	3.80	1.00	2.80	1.06	2.53	1.99	4.37
lymphoma	10.00	2.67	8.00	**0.14**	14.60	1.03	13.80	1.04	10.10	2.00	18.87
leukemia	3.20	1.07	1.20	1.00	3.20	1.10	2.20	1.05	–	–	–
prostate	4.80	1.64	3.40	1.05	4.60	1.30	4.40	0.96	–	–	–
lung	7.00	2.02	5.00	0.91	7.40	1.19	7.20	1.03	–	–	–
average	*16.42*	*2.86*	*14.48*	*0.89*	*32.90*	*1.93*	*21.07*	*1.33*	–	–	–

R = number of rules; F/R = average number of features per rule; GU = number of features globally used; CI = complexity index.

Table 9 (Cont.).

Dataset	2-SLAVE2				FeSLiC				FaIRLiC			
	R	F/R	GU	CI	R	F/R	GU	CI	R	F/R	GU	CI
korWet	14.80	7.30	44.40	2.40	10.20	4.58	25.57	0.91	6.93	2.92	14.73	**0.00**
korAgro	28.20	8.29	51.20	2.14	15.00	4.56	31.23	0.65	10.73	3.01	21.47	**0.00**
korWhole	44.00	10.19	52.80	2.19	28.97	4.94	36.83	0.57	16.60	3.82	30.27	**0.02**
thasos	15.77	7.44	87.30	2.48	8.57	3.29	27.07	0.53	6.23	2.11	12.77	**0.00**
wdcBands	9.40	6.08	49.90	2.07	9.97	2.36	21.80	0.63	9.00	2.12	17.70	**0.30**
wdcBandsSF	8.13	4.97	37.73	2.00	10.23	2.60	25.73	1.18	8.60	2.21	17.23	0.54
colon	3.40	8.53	29.40	2.18	3.10	1.46	5.00	0.11	3.57	1.79	6.50	**0.49**
dlbcl	2.90	8.91	26.53	2.16	2.80	1.64	4.90	0.29	2.73	1.30	3.80	**0.13**
lymphoma	12.40	12.75	150.77	2.52	12.57	1.82	23.00	0.73	11.40	1.95	21.90	0.48
leukemia	2.27	9.46	22.60	2.07	2.77	1.42	4.10	0.74	2.20	1.28	2.90	**0.10**
prostate	2.73	7.52	21.37	2.06	2.60	1.33	3.87	**0.03**	2.77	2.38	6.70	0.43
lung	7.27	12.75	91.13	2.95	4.90	2.56	12.63	**0.21**	7.07	2.16	15.57	1.07
average	*12.61*	*8.68*	*55.43*	*2.27*	*9.31*	*2.71*	*18.48*	*0.55*	*7.32*	*2.25*	*14.30*	***0.30***

of fuzzy sets has also been used for FeSLiC and for 2SLAVE-2, apart from the korAgro and korWhole datasets for the latter classifier, where 9 fuzzy sets per input variable (the maximum number allowed by the KEEL software) was used. The rest of the training parameters for all algorithms were set to the values proposed by their authors.

Table 7 hosts the average classification accuracy obtained by each algorithm in the training and testing sets. For three high-dimensional datasets (leukemia, prostate and lung) the SGERD algorithm could not be applied, due to excessive memory requirements (it required more than 8 GB of RAM in a 64-bit operating system). For FeSLiC and FaIRLiC, the classification accuracies after tuning are reported. Table 8 hosts the equivalent time requirement in seconds for each learning methodology. For FeSLiC and FaIRLiC, we provide the cumulative time required for the rule base extraction and the subsequent tuning process. Finally, Table 9 compares the most important structural characteristics of the various algorithms, that is, the number of rules (R), the number of features per rule (F/R) and the number of globally used (GU) features. For each dataset, we also provide a relative complexity index (CI), following a similar approach with the one proposed in Ref. 49. A deeper discussion and review of various complexity metrics can be found in Ref. 50. Considering the sth classifier, CI is calculated through:

$$\mathrm{CI}(s) = R' + F/R' + GU', \tag{23}$$

Table 10. Performance of the FeSLiC's and FaIRLiC's rule base simplification procedures for all the datasets considered.

Dataset	Simpl.?	FeSLiC				FaIRLiC			
		R	F/R	GU	L/V	R	F/R	GU	L/V
korWet	No	10.63	6.35	32.07	2.45	6.93	3.22	15.90	2.35
	Yes	10.20	4.58	25.57	1.74	6.93	2.92	14.73	1.78
korAgro	No	15.17	6.40	38.57	3.33	10.73	3.83	25.87	2.91
	Yes	15.00	4.56	31.23	2.49	10.73	3.01	21.47	2.14
korWhole	No	29.37	6.94	42.43	3.06	16.60	4.92	34.13	2.76
	Yes	28.97	4.94	36.83	2.09	16.60	3.82	30.27	1.88
thasos	No	8.83	6.95	54.83	2.64	6.23	2.88	17.10	2.44
	Yes	8.57	3.29	27.07	1.75	6.23	2.11	12.77	1.60
dcBands	No	10.67	6.88	60.80	2.59	9.00	2.94	24.07	2.03
	Yes	9.97	2.36	21.80	1.67	9.00	2.12	17.70	1.19
dcBandsSF	No	11.07	6.25	62.50	2.42	8.60	2.93	22.93	2.12
	Yes	10.23	2.60	25.73	1.80	8.60	2.21	17.23	1.32
colon	No	3.67	6.79	29.80	1.81	3.57	2.31	8.43	2.86
	Yes	3.10	1.46	5.00	1.41	3.57	1.79	6.50	2.06
dlbcl	No	3.33	25.00	87.50	1.69	2.73	1.70	5.07	2.69
	Yes	2.80	1.64	4.90	1.41	2.73	1.30	3.80	2.25
lymphoma	No	14.50	26.61	373.70	1.49	11.40	2.71	29.23	3.07
	Yes	12.57	1.82	23.00	1.19	11.40	1.95	21.90	1.65
leukemia	No	3.53	29.16	111.30	1.57	2.20	1.69	3.73	2.15
	Yes	2.77	1.42	4.10	1.28	2.20	1.28	2.90	1.25
prostate	No	3.23	49.58	207.00	1.35	2.77	3.81	10.67	2.39
	Yes	2.60	1.33	3.87	1.18	2.77	2.38	6.70	1.57
lung	No	5.63	123.33	671.30	1.32	7.07	3.12	22.33	3.41
	Yes	4.90	2.56	12.63	1.17	7.07	2.16	15.57	2.45

R = number of rules; F/R = average number of features per rule; GU = number of features globally used; L/V = number of fuzzy labels per variable.

with X' representing the normalized value of the X measure, obtained through:

$$X' = \frac{X(s) - X_{\min}}{X_{\max} - X_{\min}}, \quad (24)$$

where $X(s)$ is the value obtained for the sth classifier and $X_{\min}$ and $X_{\max}$ are, respectively, the minimum and maximum values obtained by all classifiers tested. Values of CI close to zero denote small complexity, whereas the maximum relative complexity is 3. Because the CI will be used in the statistical analysis below, SGERD was disregarded from the calculation of CI, since it cannot be executed in the last three datasets. Nevertheless, its structural characteristics in the remaining datasets are included in Table 9, for reasons of completeness.

Considering Table 7, FaIRLiC outperforms the 2SLAVE-2 and SGERD systems in all datasets, with the exception of the lymphoma dataset, for which SGERD attained higher testing accuracy. For low-dimensional datasets, FaIRLiC's classification accuracy is inferior to those of FeSLiC and the crisp classifier. Therefore, when the number of features is relatively small, the combined search for the relevant features and the antecedent part's descriptors (performed by FeSLiC and the classical rule induction algorithms) seems to be more effective than FaIRLiC's decomposed search process. Nevertheless, FaIRLiC's rule-base complexity (see Table 9) is considerably smaller in these cases. For medium-sized problems, FaIRLiC achieves equivalent classification performance, still maintaining the overall simplest structure. For high-dimensional datasets, however, FaIRLiC outperforms all other methods, except for the lymphoma and lung datasets, where C4.5Rules achieved the highest classification accuracy. In particular, the difference in classification accuracy with FeSLiC ranges from 3.31% to 17.77%, a fact that proves that for (very) high-dimensional feature spaces the search capabilities of FeSLiC's GA-based rule extraction algorithm degrade significantly. Moreover, FaIRLiC seems to be more robust against overfitting, a fact that can be deduced by the smaller difference

Table 11. Wilcoxon's test with respect to testing accuracy, p=0.05. FaIRLiC is the control algorithm.

FaIRLiC vs.	R^+	R^-	Critical value	p-value	Sig. dif.?
C4.5Rules	51	27	14	≥ 0.2	No
Ripper	60	18	14	0.10986	No
2SLAVE-2	78	0	14	4.88E-04	Yes
FeSLiC	57	21	14	0.17626	No

Table 12. Wilcoxon's test with respect to CI, p=0.05. FaIRLiC is the control algorithm.

FaIRLiC vs.	R^+	R^-	Critical value	p-value	Sig. dif.?
C4.5Rules	68	10	14	0.021	Yes
Ripper	77	1	14	9.77E-04	Yes
2SLAVE-2	78	0	14	4.88E-04	Yes
FeSLiC	58	20	14	0.15136	No

Table 13. Wilcoxon's test for FaIRLiC vs. FeSLiC, using only the medium and high-dimensional datasets, p=0.05.

	R^+	R^-	Critical value	p-value	Sig. dif.?
Classification Accuracy	42	3	6	0.019532	Yes
CI	27	18	6	≥ 0.2	No

between the training and testing accuracies, as compared to the other classifiers.

Comparing the time requirements of the considered classifiers (Table 8), we can conclude that FaIRLiC runs approximately five times faster than FeSLiC. Therefore, the decomposition of the feature selection and linguistic labels determination steps has considerably decreased the computational complexity of the algorithm, which was actually our initial motivation for the current proposal. The 2SLAVE-2 algorithm is the overall most computationally demanding classifier, whereas the SGERD algorithm – although it is fast for small and medium feature sizes –scales badly with the increase of features, at least as far as we can conclude by the three high-dimensional problems for which it was possible to be executed. In spite of the GA-based processes of FaIRLiC's learning algorithm (which typically increase significantly the computational requirements of GFRBCSs), its execution time is of the same magnitude with that of the crisp rule-base learners, which execute 2–2.5 times faster.

From the complexity point of view (Table 9), FaIRLiC achieved the lowest CI in most datasets. For the first 9 datasets, its complexity seems to be slightly inferior to SGERD's one. 2SLAVE-2 clearly produced the most complex rule-bases, whereas the crisp rule learners display medium CI values, with however a significantly higher number of rules. In particular, the number of rules for these classifiers seems to be negatively affected by large pattern sizes and/or strong overlapping between class signatures, as it is the case for the first four datasets. Compared to FaIRLiC, FeSLiC is shown to produce systems with comparable complexity. However, the average complexity measures shown in Table 9 are biased by the fact that the latter classifier produces overly simple rule-bases for high-dimensional classification tasks. If we observe its classification accuracy (Table 7) for these dataset, we can deduce that FeSLiC fails to identify the most informative features, thus restraining in simpler but suboptimal solutions. To validate this argument, we compare in Table 10 the performance of the deterministic simplification stage for both algorithms, similarly to the procedure followed for Table 3. FaIRLiC seems to be consistent in its use of the simplification stage, irrespective of the dimensionality of the feature space. FeSLiC on the other hand, while it uses this stage conservatively for low to medium feature spaces, it relies on the simplification stage to an extortionate degree for high-dimensional datasets, in order to correct the inability of the rule extraction algorithm to locate the minimum number of informative features. For example, the average number features per

rule in the case of the lung dataset are approximately 48 times more before the simplification procedure.

To validate FaIRLiC's performance, we have also conducted a statistical analysis of the results using a non-parametric statistical test, namely, the Wilcoxon's matched-pairs signed-ranks test[51–52]. The SGERD algorithm has been excluded from this analysis, because it could not be executed in the last three datasets. Table 11 hosts the results of the Wilcoxon's test with respect to the classification accuracy in the testing set. FaIRLiC is found to be statistically different with only the 2SLAVE-2 algorithm. However, Table 12 reveals that significant differences in favor of FaIRLiC exist with respect to the complexity (CI), when compared to the C4.5Rules, Ripper and 2SLAVE-2 systems. As mentioned previously, FaIRLiC performs worse than FeSLiC for low-dimensional problems. Therefore, the two systems do not show any significant statistical differences in classification accuracy. However, if we apply the test excluding the low-dimensional classification problems (Table 13), the results indicate that indeed FaIRLiC outperforms FeSLiC in medium to large feature spaces, with an associated p-value less than 0.02.

6. Conclusion

This paper presented a novel GFRBCS, namely FaIRLiC, designed under the principles of the IRL methodology. The main idea behind the proposed method is to perform feature selection and linguistic terms selection in two independent steps in the REA, contrarily to the traditional approach of handling these two objectives simultaneously. As a consequence of this divide and conquer approach, each step can handle its single objective more efficiently, resulting in compact rule bases at reduced computational costs. Comparative results using 12 real-world classification datasets proved the efficiency of the proposed methodology in significantly reducing the structural complexity of the resulting classifier, as well as its learning algorithm's computational requirements, allowing the creation of compact yet high-performing fuzzy rule bases even for very high-dimensional classification tasks.

The main disadvantage of FaIRLiC is that the user must select the appropriate number of fuzzy labels per input variable. Although this choice is necessary for other GFRBCSs, it usually entails a time-consuming trail-and-error process, until the correct number of fuzzy sets has been decided. As further research, we will try to eliminate this necessity, by introducing a hierarchical structure in the rule base, whereby rules defined in different input space granularities may coexist in the same rule base. Hence, the number of fuzzy sets will be determined by the algorithm, in a per rule basis.

References

1. M. Govender, K. Chetty, V. Naiken and H. Bulcock, A comparison of satellite hyperspectral and multispectral remote sensing imagery for improved classification and mapping of vegetation, *Water SA* **34**(2) (2008), pp. 147–154.
2. D. G. Goodenough, A. Dyk, K. O. Niemann, J. S. Pearlman, Hao Chen, T. Han, M. Murdoch and C. West, Processing Hyperion and ALI for forest classification, *IEEE Trans. Geosci. Remote Sens.* **41**(6) (2003), pp. 1321–1331.
3. H. Yang, F. V. D. Meer, W. Bakker and Z. J. Tan, A back-propagation neural network for mineralogical mapping from AVIRIS data, *Int. J. Remote Sens.* **20**(1) (1999), pp. 97–110.
4. S. Saha and S. Bandyopadhyay, A new point symmetry based fuzzy genetic clustering technique for automatic evolution of clusters, *Inf. Sci.* **179**(19) (2009), pp. 3230–3246.
5. M. A. Friedl, C. E. Brodley and A. H. Strahler, Maximizing land cover classification accuracies produced by decision trees at continental to global scales, *IEEE Trans. Geosci. Remote Sens.* **37**(2) (1999), pp. 969–977.
6. G. Camps-Valls and L. Bruzzone, Kernel-based methods for hyperspectral image classification, *IEEE Trans. Geosci. Remote Sens.* **43**(6) (2005), pp. 1351–1362.
7. N. E. Mitrakis, C. A. Topaloglou, T. K. Alexandridis, J. B. Theocharis and G. C. Zalidis, Decision Fusion of GA Self-Organizing Neuro-Fuzzy Multilayered Classifiers for Land Cover Classification Using Textural and Spectral Features, *IEEE Trans. Geosci. Remote Sens.* **46**(7) (2008), pp. 2137–2152.
8. J. A. Benediktsson and J. R. Sveinsson, Multisource remote sensing data classification based on consensus and pruning, *IEEE Trans. Geosci. Remote Sens.* **41**(4) (2003), pp. 932– 936.
9. Y. Tarabalka, J. A. Benediktsson and J. Chanussot, Spectral–Spatial Classification of Hyperspectral Imagery Based on Partitional Clustering Techniques, *IEEE Trans. Geosci. Remote Sens.* **47**(8) (2009), pp. 2973–2987.
10. X. Huang and L. Zhang, An Adaptive Mean-Shift Analysis Approach for Object Extraction and Classification From Urban Hyperspectral Imagery, *IEEE Trans. Geosci. Remote Sens.* **46**(12) (2008), pp. 4173–4185.
11. M. H. Tseng, S. J. Chen, G. H. Hwang and M. Y. Shen, A genetic algorithm rule-based approach for land-cover

classification, *ISPRS-J. Photogramm. Remote Sens.* **63**(2) (2008), pp. 202–212.
12. M. Hansen, R. Dubayah and R. Defries, Classification trees: an alternative to traditional land cover classifiers, *Int. J. Remote Sens.* **17**(5) (1996), pp. 1075–1081.
13. P. K. Goel, S. O. Prasher, R. M. Patel, J. A. Landry, R. B. Bonnell and A. A. Viau, Classification of hyperspectral data by decision trees and artificial neural networks to identify weed stress and nitrogen status of corn, *Comput. Electron. Agric.* **39**(2) (2003), pp. 67–93.
14. M. Pal and G. M. Foody, Feature selection for classification of hyperspectral data by SVM, *IEEE Trans. Geosci. Remote Sens.* **48**(5) (2010), pp.2297–2307.
15. C. Vaiphasa, A. K. Skidmore, W. F. de Boer and T. Vaiphasa, A hyperspectral band selector for plant species discrimination, *ISPRS-J. Photogramm. Remote Sens.* **62**(3) (2007), pp. 225–235.
16. A. Bárdossy and L. Samaniego, Fuzzy rule-based classification of remotely sensed imagery, *IEEE Trans. Geosci. Remote Sens.* **40**(2) (2002), pp. 362–374.
17. A. Ghosh, N. R. Pal and J. Das, A fuzzy rule based approach to cloud estimation, *Remote Sens. Environ.* **100**(4) (2006), pp. 531–549.
18. F. Melgani, B. A. R. Al Hashemy and S. M. R. Taha, An explicit fuzzy supervised classification method for multispectral remote sensing images, *IEEE Trans. Geosci. Remote Sens.* **38**(1) (2000), pp. 287–295.
19. J. H. Holland, *Adaptation in Natural and Artificial Systems* (Ann Arbor, MI: Univ. of Michigan Press, 1975).
20. O. Cordón, F. Herrera, F. Hoffmann and L. Magdalena, *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases* (Singapore: World Scientific, 2001).
21. D. G. Stavrakoudis, G. N. Galidaki, I. Z. Gitas and J. B. Theocharis, A Genetic Fuzzy-Rule-Based Classifier for Land Cover Classification From Hyperspectral Imagery, *IEEE Trans. Geosci. Remote Sens.* **50**(1) (2011), pp. 130–148.
22. H. Ishibuchi, T. Nakashima and T. Murata, Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems, *IEEE Trans. Syst. Man Cybern. Part B-Cybern.* **29**(5) (1999), pp. 601–618.
23. H. Ishibuchi, T. Nakashima and T. Morisawa, Voting in fuzzy rule-based systems for pattern classification problems, *Fuzzy Sets Syst.* **103**(2) (1999), pp. 223–238.
24. O. Cordón, M. J. del Jesus and F. Herrera, Analyzing the Reasoning Mechanisms in Fuzzy Rule-Based Classification Systems, *Mathware Soft Comput.* **5**(2–3) (1998), pp. 321–332.
25. H. Ishibuchi, T. Nakashima and M. Nii, *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining* (Springer, 2004).
26. O. Cordón, F. Gomide, F. Herrera, F. Hoffmann and L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends, *Fuzzy Sets Syst.* **141**(1) (2004), pp. 5–31.
27. J. Casillas, P. Martínez and A.D. Benítez, Learning consistent, complete and compact sets of fuzzy rules in conjunctive normal form for regression problems, *Soft Comput.* **13**(5) (2009), pp. 451–465.
28. D. P. Greene and S. F. Smith, Competition-based induction of decision models from examples, *Mach. Learn.* **13**(2–3) (1993), pp. 229–257.
29. O. Cordón, M.J. del Jesús, F. Herrera and 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), pp. 1123–1153.
30. A. González and R. Pérez, SLAVE: a genetic learning system based on an iterative approach, *IEEE Trans. Fuzzy Syst.* **7**(2) (1999), pp. 176–191.
31. A. González and R. Pérez, Selection of relevant features in a fuzzy genetic learning algorithm, *IEEE Trans. Syst. Man Cybern. Part B-Cybern.* **31**(3) (2001), pp. 417–425.
32. F. Hoffmann, Combining boosting and evolutionary algorithms for learning of fuzzy classification rules, *Fuzzy Sets Syst.* **141**(1) (2004), pp. 47–58.
33. M.J. del Jesus, F. Hoffmann, L.J. Navascués and L. Sánchez, Induction of Fuzzy-Rule-Based Classifiers With Evolutionary Boosting Algorithms, *IEEE Trans. Fuzzy Syst.* **12**(3) (2004), pp. 296–308.
34. Y. Freund and R. Schapire, Experiments with a new boosting algorithm, in *Proc. 13th Int. Conf. Machine Learning* (1996), pp. 148–156.
35. D. G. Stavrakoudis, J. B. Theocharis and G. C. Zalidis, A multistage genetic fuzzy classifier for land cover classification from satellite imagery, *Soft Comput.* **15**(12) (2011), pp. 2355–2374.
36. S. P. Moustakidis, J. B. Theocharis and G. Giakas, Subject recognition based on ground reaction force measurements of gait signals, *IEEE Trans. Syst. Man Cybern. Part B-Cybern.* **38**(6) (2008), pp. 1476–1485.
37. T. Bäck, *Evolutionary Algorithms in Theory and Practice* (Oxford University Press, Oxford, 1996).
38. G. H. Mitri and I. Z. Gitas, Mapping Postfire Vegetation Recovery Using EO-1 Hyperion Imagery, *IEEE Trans. Geosci. Remote Sens.* **48**(3) (2010), pp. 1613–1618.
39. S. G. Ungar, J. S. Pearlman, J. A. Mendenhall and D. Reuter, Overview of the Earth Observing One (EO-1) mission, *IEEE Trans. Geosci. Remote Sens.* **41**(6) (2003), pp. 1149–1159.
40. 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), pp. 359–365.
41. E. G. Mansoori, M. J. Zolghadri and S. D. Katebi, SGERD: A Steady-State Genetic Algorithm for Extracting Fuzzy Classification Rules From Data, *IEEE Trans. Fuzzy Syst.* **16**(4) (2008), pp. 1061–1071.
42. J. Alcalá-Fdez, L. Sánchez, S. García, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández and F. Herrera, KEEL: A

Software Tool to Assess Evolutionary Algorithms to Data Mining Problems, *Soft Comput.* **13**(3) (2009), pp. 307–318. Software available online: http://www.keel.es.

43. J. R. Quinlan, *C4.5: Programs for Machine Learning* (Morgan Kauffman, 1993).
44. D. G. Stavrakoudis, J. B. Theocharis and G. C. Zalidis, A Boosted Genetic Fuzzy Classifier for land cover classification of remote sensing imagery, *ISPRS-J. Photogramm. Remote Sens.* **66**(4) (2011), pp. 529–544.
45. D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing* (Hoboken, NJ: Wiley, 2003).
46. X. Huang, L. Zhang and P. Li, Classification and Extraction of Spatial Features in Urban Areas Using High-Resolution Multispectral Imagery, *IEEE Geosci. Remote Sens. Lett.* **4**(2) (2007), pp. 260–264.
47. S. P. Moustakidis and J. B. Theocharis, SVM-FuzCoC: A novel SVM-based feature selection method using a fuzzy complementary criterion, *Pattern Recogn.* **43**(11) (2010), pp. 3712–3729.
48. W. W. Cohen, Fast Effective Rule Induction, in *Proc. 12th Int. Conf. Machine Learning (ML95)*, California (1995).
49. F. J. Berlanga, A. J. Rivera, M. J. del Jesus and 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), pp. 1183-1200.
50. M. J. Gacto, R. Alcala and F. Herrera, Interpretability of Linguistic Fuzzy Rule-Based Systems: An Overview of Interpretability Measures, *Inf. Sci.* (in press), DOI: 10.1016/j.ins.2011.02.021.
51. J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.* **7** (2006), pp. 1–30.
52. S. García, D. Molina, M. Lozano and 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, *J. Heuristics* **15**(6) (2009), pp. 617–644.