

A Hierarchical Genetic Fuzzy Rule-Based Classifier for High-Dimensional Classification Problems

Dimitris G. Stavrakoudis^a, Ioannis Z. Gitas^b, and John B. Theocharis^a

^aDepartment of Electrical and Computer Engineering

^bSchool of Forestry and Natural Environment

Aristotle University of Thessaloniki

Thessaloniki, Greece

{jstavrak@auth.gr; igitas@for.auth.gr; theochar@eng.auth.gr}

Abstract—This paper proposes a novel Hierarchical Genetic Fuzzy Rule-Based Classification System (HGFRBCS), targeted at effectively handling high-dimensional classification tasks. A hierarchical fuzzy rule base comprises rules with linguistic terms from a multi-granular fuzzy sets database, whereby lower levels define thicker granularities of the input space fuzzy partition. The proposed system is developed through sequential repeating steps: in each step a fuzzy rule base is created using a given granularity. Subsequently, the best performing rules are inserted in the hierarchical rule base and the process is repeated again, considering a thicker granularity. The whole process is coordinated by a boosting scheme, which localizes new rules in uncovered regions of the feature space. Comparative results for various real-world high-dimensional classification problems indicate the effectiveness of the proposed methodology.

Fuzzy rule-based classification system (FRBCS); hierarchical fuzzy partitions; genetic fuzzy systems

I. INTRODUCTION

The interpretational properties of Genetic Fuzzy Rule-Based Classification Systems (GFRBCSs) [1] have rendered them useful tools in the field of data mining over the past two decades [2]. In particular, linguistic FRBCSs [3]–[4] have the most intuitive representation, comprising rules with linguistic terms from a common fuzzy sets database. Their main drawback is a possible loss in accuracy when dealing with complex classification tasks, resulting from the inflexibility introduced by the discretization of the input space into a fixed number of fuzzy sets with predefined membership functions.

In order to increase classification accuracy, a hierarchical framework for building FRBCSs has been recently proposed [5]–[6]. This approach applies a two-level methodology. In the first level, a fuzzy rule base is created with an initial fuzzy set database. Poorly performing rules are removed from the rule base and replaced by rules defined in a thicker fuzzy partition granularity. In the second level, a genetic rule selection procedure is performed, which selects the best cooperating rules from the hierarchical rule base (HRB). However, the rule extraction algorithm (REA) considered in this approach results in the generation of a high number of fuzzy rules, as the results in [6] indicate. Another drawback of the approach is that it does not consider any feature reduction procedure, resulting in fuzzy

rules with a complex linguistic structure when dealing with high-dimensional classification problems.

Another HGFRBCS has been previously proposed in [7], whose HRB structure resembles the philosophy followed by the herein proposed system. In this approach, an initial pool of candidate fuzzy rules is considered, defined in fuzzy partitions with different granularities. Subsequently, a genetic rule selection procedure selects the best cooperating rules, in order to form the final HRB. The main drawback of this approach lies in the creation of the initial pool of fuzzy rules, which is performed by considering the complete rule base for each fuzzy partition granularity, rendering the procedure ineffective in high-dimensional problems.

In this paper we propose a HGFRBCS which aims at producing compact yet high-performing HRBs in high-dimensional classification tasks. The basic idea is to repeatedly invoke a rule base extraction algorithm (RBEA), each time considering a fuzzy partition of thicker granularity. Fuzzy rules that increase the HRB's performance are maintained, whereas the rest are disregarded. In order for new rules to be created in uncovered regions of the feature space, we consider a boosting scheme that penalizes previously well covered patterns, so that new valuable rules will be localized in uncovered or mixed regions of the feature space. As the basic RBEA we consider a GFRBCS from a previous contribution of ours [8], which has been proven to generate compact rule bases, with low computational requirements.

The rest of the paper is organized as follows. Section II summarizes some of the basic concepts that will be used throughout the paper. In Section III we provide a brief description of the basic RBEA. Section IV details the framework for building the hierarchical fuzzy rule base. Experimental results for a number of high-dimensional classification problems are reported in Section V and the paper concludes in Section VI with some final remarks.

II. PRELIMINARIES

This section introduces the notation that will be used in the rest of the paper, along with a description of the proposed system's fuzzy inference mechanism implementation and the boosting algorithm considered.

A. Fuzzy Reasoning

We consider a classification problem defined in an N -dimensional feature space $\mathbf{x} = [x_1, \dots, x_N] \in \mathcal{R}^N$ and an M -dimensional class space $\mathbf{C} = \{C_1, \dots, C_M\}$ and a set of Q labeled patterns $E = \{(\mathbf{x}^p, c^p), p = 1, \dots, Q\}$. The proposed FRBCS comprises a number of linguistic DNF fuzzy rules, having the form:

$$R_k: \text{IF } x_1 \text{ is } A_1^k \text{ and } \dots x_N \text{ is } A_N^k \text{ THEN } y \text{ is } C^k \text{ with } r^k, \quad (1)$$

where each input variable x_i takes as a value a set of linguistic terms $A_i^k = \{L_i^1 \text{ or } \dots \text{ or } L_i^{\ell_k}\}$ joined by a disjunctive operator, whereas the output variable (y) is one of the class labels $C^k \in \mathbf{C}$. We will also consider that some input variables in each rule may 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. Moreover, each fuzzy rule includes a certainty degree $r^k \in [0, 1]$ which represents the confidence of the classification in the class label of the consequent (C^k). This certainty degree is calculated in our proposal through

$$r^k = \sum_{p|c^p=C^k} \mu^k(\mathbf{x}^p) / \sum_{p=1}^Q \mu^k(\mathbf{x}^p), \quad (2)$$

where $\mu^k(\mathbf{x}^p)$ is the degree of compatibility between an example and the antecedent part of a fuzzy rule (also known as matching degree). This matching degree is calculated as follows:

$$\mu^k(\mathbf{x}^p) = \bigwedge_{i=1}^N \left\{ \bigvee_{q=1}^{\ell_k} \mu_{L_i^q}(x_i^p) \right\}, \quad (3)$$

where $\mu_{L_i^q}(x_i^p)$ is the membership grade of each linguistic term participating in the formulation of i th 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, whereas the OR operator is implemented through the bounded sum, defined for two membership values a and b as:

$$\text{bs}(a, b) = \min(1, a + b) \quad (4)$$

With this choice, 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.

Since the proposed system implements a HRB, the linguistic terms A_i^k in each rule are defined in a fuzzy partition of a certain granularity. Hence, a hierarchical database (HDB) of fuzzy sets is defined, comprising S hierarchy levels with

respective granularities $G(t)$, $t = 1, \dots, S$. All linguistic terms participating in one rule are restricted to belong to the same hierarchy level.

Finally, we use the so-called maximum voting scheme in order to classify a given feature vector $\mathbf{x}^p$, defined as:

$$C_{\max} = \arg \max_{j=1, \dots, M} \sum_{R^k | C^k = j} \mu^k(\mathbf{x}^p) \cdot r^k \quad (5)$$

B. Boosting Scheme

As mentioned in the introduction, the proposed system iteratively calls a RBEA, considering each time a fuzzy partition of thicker granularity. In order to localize new rules in uncovered regions of the feature space, we need a mechanism which will penalize previously covered examples. Such a framework can be provided by the so-called boosting algorithms. AdaBoost [9] is the most well-known boosting algorithm and its main idea is to combine a set of low quality classifiers (called weak hypotheses in the boosting literature) with a voting scheme, in order to generate an overall classifier that performs better than any of its constituents alone. In the context of FRBCSs, each fuzzy rule can be regarded as a simple but partial (in terms of overall classification performance) classifier. Assuming a training patterns set $E_{\text{trn}} = \{e^p = (\mathbf{x}^p, c^p), p = 1, \dots, Q\}$, the AdaBoost algorithm assigns a weight w^p to each one of the Q training patterns, all initialized with a unity value $w^p = 1$. Each time a new rule R^k is inserted in the HRB, AdaBoost computes the error $E(R^k)$ of the rule, taking into consideration the examples’ previous weights and their matching degrees:

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

Incorrectly classified patterns retain their former weights, whereas the weights of correctly classified patterns are reduced by a factor $\beta^k(\mathbf{x}^p)$:

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

The factor

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

depends on the current rule’s error and its matching degree with the p th pattern. Effectively, examples that are classified correctly and match the rule antecedent are down-weighted, whereas misclassified or uncovered examples maintain their original weights.

We should note that there is an important difference between fuzzy rules and the notion of the weak hypothesis in the boosting literature. The weak hypothesis is assumed to be a low performing classifier that, however, covers the whole feature space. On the contrary, a fuzzy rule is a local model, covering only a specific subspace of the feature space.

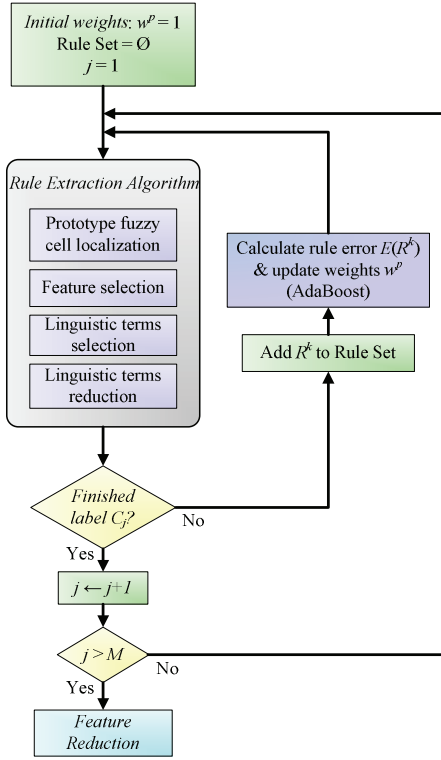


Figure 1. Outline of FaIRLiC's learning algorithm.

Therefore, under the AdaBoost philosophy, classifiers that are created at later stages handle mostly uncovered regions of the feature space, but are allowed to also (correctly) classify previously down-weighted patterns. In FRBCSs this is generally an undesirable behavior, since superfluous fuzzy rules with high overlapping may be produced. To that extent, special provision must be made to avoid such phenomena when applying the AdaBoost algorithm in the derivation of GFRBCSs.

III. RULE BASE EXTRACTION ALGORITHM

In a previous work [8], we introduced the so-called Fast Iterative Rule-Based Linguistic Classifier (FaIRLiC), a GFRBCS designed under the Iterative Rule Learning (IRL) approach [1]. This system is characterized by the compact rule base structure it produces, the overall good classification accuracy attained and the relatively low time requirements of its learning algorithm. Due to these attributes it has been selected here as the basic RBEA of the proposed system. In the rest of this section, we provide a brief description of FaIRLiC. A more comprehensive description can be found in the aforementioned reference.

An outline of FaIRLiC's learning algorithm is depicted in Fig. 1. FaIRLiC constructs a FRBCS comprising DNF rules in the form of (1), with a fuzzy partition of a certain granularity, selected by the designer. Following the IRL approach, the system is constructed through repeated invocation to a rule extraction algorithm, each time returning the best fuzzy rule found under the current distribution of the training examples. This distribution is controlled by the AdaBoost algorithm

presented in Section II.B. Hence, each time a new rule is added to the final rule base, the weights of correctly covered examples are down-weighted and subsequent invocation to the algorithm concentrate on uncovered patterns.

The rule extraction algorithm is performed in four successive steps: a) prototype fuzzy cell localization, b) feature selection, c) linguistic terms selection and d) linguistic terms reduction. The first step initializes the algorithm by selecting the most promising single linguistic term for each input variable. Given these terms, the second step performs a genetic feature selection procedure. After having selected the relevant features of the rule, a genetic algorithm (GA) in the third step identifies the DNF-type linguistic terms that will be included in its antecedent part, for each active feature. The final step is a simple deterministic one, which corrects the fuzzy rule from extending to empty subspaces of the feature space. Specifically, for each active input variable we remove one fuzzy term at a time (if there are more than two) and accept that change if the classification performance 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.

Throughout FaIRLiC's rule extraction algorithm we evaluate extensively a feature selection oriented metric, the so-called Feature Partition Vector (FPV). The FPV quantifies the degree to which each feature can correctly classify each one of the training examples, independently from other features. This metric is used in order to localize the initial linguistic terms (first step) in the most informative region of the equivalent input variable, as well as for the initialization of the population and the application of special genetic operators in the second step. Moreover, a modified coverage fitness function is considered for the GAs in the second and third steps, which explicitly discourages new rules to cover previously well-covered examples.

Similarly to the SLAVE algorithm [10], we learn one class label at a time, with no particular order being assumed. When no more useful rules can be produced for the class under consideration, the algorithm proceeds to the next class label. Two conditions are considered to judge if the algorithm should proceed to the next class label:

- The new rule's fitness value is zero (in either the feature selection or the linguistic terms selection steps), which signifies the inability of the rule extraction algorithm to find a new useful rule.
- The classification performance of the rule base obtained so far does not increase more than a pre-specified threshold on a validation set of examples. Accordingly, our initial set of labeled patterns is split into three disjoint sets: the training, the validation, and the testing ones.

As a final step, we perform a deterministic feature reduction step, which is similar to the linguistic terms reduction one. Specifically, 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

Initialization: $w^p = 1$; $a^{NPR} = a_{\text{init}}^{NPR}$; $\text{HRB} = \emptyset$

```

for hierarchy level  $t = 1$  to  $S$ 
  call FaIRLiC( $G(t)$ ,  $w^p$ )
  find promising rules set  $R_{pr}(t)$ 
  for  $k = 1$  to  $|R_{pr}(t)|$ 
    insert  $k$ th rule into HRB
    if validation performance increases
      update weights  $w^p$ 
    else
      remove  $k$ th rule from HRB
    endif
  endfor
  adjust  $a^{NPR}$ 
endfor
Perform rule reduction
Perform feature reduction

```

Figure 2. An outline of the proposed methodology.

the chance. Moreover, since removing a feature results in a generalization of the rule, we also assure that the overlapping between rules describing the same class label does not increase.

IV. ITERATIVE HIERARCHICAL FRAMEWORK

Here we introduce the Hierarchical Rule-Based Linguistic Classifier (HiRLiC), which is a framework for building HGFRBCSs. As mentioned in the introduction, HiRLiC repeatedly calls a RBEA (the FaIRLiC algorithm), each time considering a fuzzy partition of thicker granularity. Since FaIRLiC considers DNF-type rules in the form of (1), one might suggest to select a very fine granularity of the input space and then request FaIRLiC to implicitly identify the required granularity for each rule, by selecting the appropriate linguistic terms to be joined by the OR disjunctive operator. However, in an N -dimensional classification problem with each input variable being partitioned into ℓ fuzzy sets, plus the “don’t care” (inactive feature) fuzzy set, the search space for simple linguistic rules is $(\ell + 1)^N$. The actual search space is much larger if we consider the DNF type of the fuzzy rules. As the dimensionality of the feature space increases, the search space of the problem increases exponentially, thus degrading the effectiveness of FaIRLiC’s GA-driven learning algorithm.

To overcome the aforementioned problem, HiRLiC decomposes the problem into smaller, easier to tackle sub-problems. Through AdaBoost, the algorithm initially concentrates in uncovered regions of the feature space, allowing the mixed areas in the borders of neighboring classes to be handled in a later stage, when a finer partition will be available and most of the easily discriminated patterns will have been practically removed (their weight values will be close to zero). When seen under this scope, HiRLiC employs a higher-level IRL methodology: just as FaIRLiC repeatedly calls a rule extraction algorithm and uses the (local) AdaBoost weights to localize new rules in uncovered regions of the feature space, so does HiRLiC iteratively invoke a RBEA in different hierarchy levels, using the (global) AdaBoost weights as the control mechanism. To that extent, HiRLiC is able to effectively handle the large search space, thus increasing its base weak learner’s, that is FaIRLiC’s, classification accuracy.

TABLE I. SUMMARY DESCRIPTION OF THE DATASETS

Dataset	Patterns	Features	Classes
Ionosphere	351	34	2
Libras	360	90	15
Optdigits	5620	64	10
Satimage	6435	36	6
Sonar	208	60	2
Spambase	4601	57	2
Wdbc	569	30	2
Texture	5500	40	11
KorAgro	2706	53	8
KorWet	1219	53	5
KorWhole	3925	53	13
Thasos	1000	198	6

An outline of the HiRLiC learning methodology is given in Fig. 2. Initially we consider an empty HRB and the AdaBoost algorithm assigns unity weights to the training patterns. The designer selects the number S of the hierarchy levels to be considered, along with the respective granularities of the fuzzy partitions $G(t)$, $t = 1, \dots, S$. For each hierarchy level t , the FaIRLiC algorithm is executed, producing fuzzy rules defined in a fuzzy partition with $G(t)$ granularity, as shown in Fig. 1. The only difference is that instead of initializing FaIRLiC’s AdaBoost weights with unity values (top box in Fig. 1) we provide a copy HiRLiC’s (global) current weight values.

Obviously, the most crucial point is the selection of the rules to be inserted in the HRB and the ones that will be rejected. We cannot rely solely on the classification accuracy of the HRB, since rules describing currently unhandled class labels will always increase its classification accuracy. In particular, we want to (at least initially) promote confident rules, that is, rules covering a high number of positive examples and a low number of negative ones. Therefore, we propose the use of the negative to positive examples ratio (NPR), which is simply defined as the ratio of the fuzzy number of positive examples n^+ to the fuzzy number of negative ones n^- . For the k th fuzzy rule, describing the C^k class label, these numbers are defined, respectively, as:

$$n^+ = \sum_{p|c^p = C^k} \mu^k(\mathbf{x}^p), \quad (9)$$

$$n^- = \sum_{p|c^p \neq C^k} \mu^k(\mathbf{x}^p). \quad (10)$$

If the NPR of a rule is lower than a threshold value a^{NPR} , then this rule is considered to be a *promising* one and it is inserted into the respective rule set of promising rules $R_{pr}(t)$. Generally, we should be stricter in the initial steps, where coarse granularities are considered, and relax this restriction as the algorithm proceeds to lower levels of the hierarchy. To achieve this, the designer chooses an initial threshold value $a_{\text{init}}^{NPR} < 1$ and this threshold increases linearly for subsequent hierarchy levels, achieving a unity value in the last (S) level. Hence, the threshold value at level t is expressed as:

$$a_t^{NPR} = a_{\text{init}}^{NPR} + (1 - a_{\text{init}}^{NPR}) \frac{t-1}{S-1}. \quad (11)$$

TABLE II. RESULTS OBTAINED BY FAIRLiC

Dataset	P (%)	# R	GU	#F/R	T (s)
Ionosphere	84.80	3.63	3.97	1.43	1.40
Libras	41.02	20.33	40.63	2.62	14.58
Optdigits	75.85	12.10	27.23	3.23	38.05
Satimage	80.04	8.67	17.00	2.60	24.60
Sonar	67.80	4.43	9.87	2.21	1.98
Spambase	82.09	2.30	10.63	4.73	7.12
Wdbc	90.40	2.17	2.50	1.16	1.30
Texture	82.96	12.87	21.00	2.80	33.78
KorAgro	61.28	9.53	23.50	3.66	22.17
KorWet	76.51	7.27	13.67	2.56	7.72
KorWhole	59.02	15.30	28.70	4.05	47.33
Thasos	81.30	6.23	12.77	2.11	11.13
Average	73.59	8.74	17.62	2.76	17.60

P (%) = Testing performance (%); # R = number of rules; GU = number of features globally used; # F/R = average number of features per rule.

After creating the rule set $R_{pr}(t)$, we insert each one of the rules into the HRB (one at a time), in the order of their creation. If the classification performance of the HRB on the validation set of examples increases with the insertion of a rule, then this rule is accepted and HiRLiC's AdaBoost weights are updated (see Fig. 2), according to (6)–(8). Otherwise the rule is rejected. Through this procedure, only rules that cooperate well with the previous HRB are inserted into the final model.

Note that all available S hierarchy levels are evaluated. Nevertheless, it is acceptable that no rules with a certain granularity $G(t)$ will be inserted in the final model. Hence, the final system utilizes only the necessary levels of the hierarchical fuzzy sets database. We should also remark, that easily discriminated classes with broad coverage will be generally handled first, which in general will result in faster execution of the FaIRLiC algorithm in later steps.

Finally, before exporting the final HRB, we perform two deterministic rule base refinement procedures, namely, rule reduction and feature reduction, as shown in Fig. 2. The features reduction algorithm is the same as the one performed at the end of FaIRLiC's algorithm (see Section III), whereas rule reduction is a similar procedure which removes one rule at a time and accepts the change if both the training and validation performance rates do not degrade. Arguably, the objectives pursued by these procedures have already been achieved in previous steps to a large extent. Indeed, for the datasets considered in our experimental setup, the average decrease in the numbers of rules and features per rule was 9.74 % and 5.07 %, respectively. Nevertheless, their application is advocated by their practically negligent computational requirements.

V. EXPERIMENTAL STUDY

In order to analyze the performance of the proposed methodology we considered 12 real-world experiments with a medium to high number of features. Table I summarizes the most important characteristics of these datasets. The first 7 datasets are from the UCI repository [11], whereas the Texture dataset is from the ELENA project [12]. The last four are remote sensing classification tasks from satellite imagery. In

TABLE III. RESULTS OBTAINED BY HiRLiC

Dataset	P (%)	HD	# R	GU	#F/R	T (s)
Ionosphere	90.31	3.63	6.67	10.30	1.85	9.80
Libras	51.41	6.90	27.90	53.40	2.88	114.08
Optdigits	82.00	6.93	26.20	43.77	3.53	275.56
Satimage	83.36	6.83	17.67	26.53	2.81	212.42
Sonar	69.92	4.60	7.87	21.07	3.23	14.96
Spambase	87.82	3.50	5.10	16.30	4.36	49.54
Wdbc	93.13	2.43	3.67	5.27	1.55	9.89
Texture	86.22	6.90	22.60	26.70	2.93	306.10
KorAgro	69.35	6.53	19.90	31.23	3.35	157.22
KorWet	81.11	4.77	10.30	18.60	2.46	52.30
KorWhole	63.88	6.77	28.53	34.60	3.72	347.14
Thasos	83.60	5.63	11.40	25.13	2.53	98.53
Average	78.51	5.45	15.65	26.08	2.93	137.30

P (%) = testing performance (%); HD = hierarchy depth; # R = number of rules; GU = number of features globally used; # F/R = average number of features per rule.

particular, a multispectral IKONOS image was acquired from an area surrounding Lake Koronia in northern Greece. Advanced higher order spectral and spatial features were subsequently derived from the initial four bands of the image, 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 agricultural (KorAgro) and the wetland (KorWet) ones. For the sake of completeness, the whole (original) dataset is also considered (KorWhole) here. A detailed description of these datasets can be found in [13]–[14]. Finally, the Thasos dataset was obtained from a hyperspectral Hyperion image (level 1 radiometric product), covering a part of Thasos, Greece's most northerly island [15]. It comprises a total of 198 features in the visible near infrared (VNIR) (400–1000 nm) and shortwave infrared (900–2500 nm) range of the electromagnetic spectrum.

To develop the different experiments we consider a 5-fold cross-validation model, that is, 5 random partitions with 20 % of the data and the combination of 4 of them (80 %) as training and the remaining 20 % as testing. Due to the computational requirements of FaIRLiC and HiRLiC, the training partitions were further split into two equal parts, resulting in the final training and validation partitions. Moreover, we performed six independent runs on each partition, using different random seeds, since GAs are stochastic processes. In each case, the average results over these 30 runs are given.

The parameters used for FaIRLiC are the same as the ones proposed in [8], either when running it independently or as a part of HiRLiC's learning algorithm. When FaIRLiC is tested as a standalone classifier, we use 9 fuzzy sets per input variable, as a reasonable number of fuzzy sets used in many GFRBCSs of the respective literature. In applying HiRLiC, we consider $S = 7$ hierarchy levels with odd granularities from 3 to 15, that is, $G = \{3, 5, 7, 9, 11, 13, 15\}$. The initial NPR threshold for HiRLiC was set to $a_{init}^{NPR} = 0.2$, so that at the first iteration (3 fuzzy sets per input variable) a rule is inserted into the promising rule set only if its negative examples are fewer than 20 % of its positive ones. The two systems were coded in C++ and all experiments were conducted on an Intel Core 2 Quad Q9650 at 3.0 GHz.

TABLE IV. COMPARISON OF VARIOUS GFRBCSS

Dataset	SGERD				Ishibuchi-GCCL				2SLAVE-2				HiRLiC			
	P(%)	#R	GU	#F/R	P(%)	#R	GU	#F/R	P(%)	#R	GU	#F/R	P(%)	#R	GU	#F/R
Ionosphere	72.10	2.00	3.60	2.00	87.18	81.00	32.20	2.56	88.89	4.80	12.00	2.77	90.31	6.67	10.30	1.85
Libras	41.73	20.00	27.40	2.00	19.53	39.20	78.60	6.66	57.60	49.60	89.60	11.79	51.41	27.90	53.40	2.88
Optdigits	55.20	21.00	21.20	1.98	40.18	82.60	59.40	5.94	76.96	63.20	63.80	7.54	82.00	26.20	43.77	3.53
Satimage	77.01	12.80	14.20	2.00	74.50	71.00	35.40	2.58	84.27	53.60	36.00	10.22	83.36	17.67	26.53	2.81
Sonar	72.59	3.00	4.00	2.00	65.80	58.40	53.20	3.82	64.66	6.40	29.20	6.21	69.92	7.87	21.07	3.23
Spambase	69.72	2.40	4.20	2.00	71.65	76.20	53.40	6.94	85.74	19.40	44.80	4.51	87.82	5.10	16.30	4.36
Wdbc	91.42	3.60	5.40	2.00	91.22	59.60	26.20	1.68	93.69	5.40	14.80	3.37	93.13	3.67	5.27	1.55
Texture	69.40	16.20	16.00	2.00	66.33	80.80	36.20	2.62	87.18	33.40	40.00	9.99	86.22	22.60	26.70	2.93
KorAgro	60.98	11.23	12.83	1.95	50.92	62.60	46.80	4.02	59.14	17.30	43.50	6.60	69.35	19.90	31.23	3.35
KorWet	73.28	6.70	7.93	1.94	69.09	58.40	43.20	4.36	80.35	11.17	34.13	5.71	81.11	10.30	18.60	2.46
KorWhole	47.00	17.50	17.43	1.99	40.66	71.60	50.00	4.57	50.89	25.60	46.63	7.12	63.88	28.53	34.60	3.72
Thasos	75.21	7.50	12.37	2.00	—	—	—	—	80.18	15.77	87.30	7.44	83.60	11.40	25.13	2.53
Average	67.14	10.33	12.21	1.99	61.55	67.40	46.78	4.16	75.80	25.47	45.15	6.94	78.51	15.65	26.08	2.93

P (%) = Testing performance (%); # R = number of rules; GU = number of features globally used; # F/R = average number of features per rule.

We will first compare the average results obtained by FaIRLiC and HiRLiC, which are shown in Tables II and III, respectively. These tables include – for each dataset – the testing classification accuracy, the number of rules, the average number of globally used features, the number of features per rule, and the runtime of each algorithm in seconds. The term globally used features is used to express the number of features being evaluated by the whole rule base, that is, the features that appear at least once in any rule of the system. Table III also includes the average hierarchy depth of HiRLiC, meaning the average number of hierarchy levels that are actually being utilized by at least one rule of the rule base. The last row of each table reports the respective average values for all datasets considered.

From the performance point of view, HiRLiC outperforms FaIRLiC in all datasets, with the additional performance gain ranging from 2.12 % (Sonar) to 10.39 % (Libras). In most cases, HiRLiC's classification performance is more than 3 % higher than FaIRLiC's ones. This fact proves the fulfillment of our primary objective, which was to increase the classification ability of the base classifier. From the model simplicity point of view, the number of features per rule have marginally increased, while the rule base size increased, subsequently resulting in a higher number of globally used features. However, observing the feature and class space dimensionality of each dataset, we can argue that the obtained HRBs are generally compact, retaining their former interpretability properties to a considerable degree. Obviously, the time requirements of HiRLiC's learning algorithm is multiple of FaIRLiC's ones, since the former repeatedly invokes the latter. Nevertheless, FaIRLiC's learning is fast enough to maintain the HiRLiC's execution times to an acceptable level, especially if we take under consideration the high number of training patterns for most of the datasets.

To validate HiRLiC's performance, we compared it against three other GFRBCSs of the literature, namely, the SGERD [16], Ishibuchi's GCCL [17] and 2SLAVE-2 [10] systems, tested using the Keel software package [18]. The number of fuzzy sets per input variable for 2SLAVE-2 was set to 9, as we have done with FaIRLiC. For Ishibuchi's GCCL we tested granularities from 3 to 9 (the maximum allowed in the Keel

software) fuzzy sets per input variable and we finally chose a granularity value of 5, as it achieved the highest classification accuracy in the vast majority of the considered datasets. The rest of the parameters for these three systems were set to the respective values proposed by the authors.

In Table IV we present a comparison of HiRLiC with the three aforementioned GFRBCSs. The results for Ishibuchi's GCCL in the Thasos dataset are missing because the method was unable to produce any useful rules (zero number of rules was obtained) for this dataset. Performance-wise, HiRLiC was found to clearly outperform SGERD and Ishibuchi's GCCL, as well as 2SLAVE-2 in most cases, although the difference with the latter is smaller. However, 2SLAVE-2 resulted in significantly more complex rule bases, with a higher number of features per rules and number of rules than HiRLiC. Ishibuchi's GCCL resulted in fuzzy rule bases with high number of rules, whereas the structure of SGERD is overall approximately equivalent to FaIRLiC's one (see Table II), although the former attained lower classification accuracies in most datasets. From this comparative analysis, we can conclude that HiRLiC results in high performing rule-based classifiers, maintaining a reasonably compact rule base structure at the same time.

VI. CONCLUSION

This paper presented a novel framework for developing hierarchical GFRBCSs, targeting at increasing the classification accuracy of the obtained model. HiRLiC is constructed through an iterative process, by repeated invocations to a basic RBEA, each time considering a fuzzy partition of thicker granularity. A boosting algorithm is responsible for focusing the algorithm towards the uncovered or highly mixed regions of the feature space. The rules provided by the RBEA are evaluated by the HRB and are included in the final model only if they are confident and increase the system's performance. Extensive experimental analysis on various real-world high-dimensional classification tasks has proved the proposed methodology's ability to increase the classification accuracy of the system, maintaining at the same time a reasonably compact structure.

REFERENCES

- [1] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. Singapore: World Scientific, 2001.
- [2] 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.*, vol. 141, no. 1, pp. 5–31, 2004.
- [3] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*. Springer, 2004.
- [4] M. J. Gacto, R. Alcalá, and F. Herrera, "Interpretability of Linguistic Fuzzy Rule-Based Systems: An Overview of Interpretability Measures." *Inf. Sci.*, in press, 2011. DOI: 10.1016/j.ins.2011.02.021.
- [5] A. Fernandez, M. J. del Jesus, and F. Herrera, "Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets," *Int. J. Approx. Reasoning*, vol. 50, no. 3, pp. 561–577, 2009.
- [6] A. Fernandez, M. J. del Jesus, and F. Herrera, "Analysing the Hierarchical Fuzzy Rule Based Classification Systems with Genetic Rule Selection," *4th Int. Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS)*, pp. 69–74, 2010.
- [7] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 260–270, 1995.
- [8] D. Stavrakoudis, G. Galidaki, I. Gitas, and J. Theoharis, "A Fast Iterative Rule-Based Linguistic Classifier for Hyperspectral Remote Sensing Tasks", *Proc. 5th Int. Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS)*, pp. 25–30, 2011.
- [9] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Machine Learning*, pp. 148–156, 1996.
- [10] A. González and R. Pérez, "Selection of relevant features in a fuzzy genetic learning algorithm," *IEEE Trans. Syst. Man Cybern., Part B-Cybern.*, vol. 31, no. 3, pp. 417–425, 2001.
- [11] A. Asuncion and D. Newman, "UCI machine learning repository," Irvine, CA: Dept. Inf. Comput. Sci., Univ. California, 2007. [Online]. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [12] C. Aviles-Cruz, A. Gurin-Dugu, J. L. Voz, and D. Van Cappel, "Databases, Enhanced Learning for Evolutive Neural Architecture," Tech. Rep. R3-B4-P, 1995. [Online]. Available: <http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/elena.htm>.
- [13] D. G. Stavrakoudis, J. B. Theoharis, and G. C. Zalidis, "A multistage genetic fuzzy classifier for land cover classification from satellite imagery," *Soft Comput.*, in press, 2011. DOI: 10.1007/s00500-010-0666-z.
- [14] D. G. Stavrakoudis, J. B. Theoharis, and G. C. Zalidis, "A Boosted Genetic Fuzzy Classifier for land cover classification of remote sensing imagery," *ISPRS-J. Photogramm. Remote Sens.*, in press, 2011. DOI: 10.1016/j.isprsjprs.2011.01.010.
- [15] D. G. Stavrakoudis, G. N. Galidaki, I. Z. Gitas, and J. B. Theoharis, "Enhancing the interpretability of genetic fuzzy classifiers in land cover classification from hyperspectral satellite imagery," in *2010 Int. Conf. Fuzzy Systems (IEEE-FUZZ)*, Barcelona, pp. 1277–1284, 2010.
- [16] 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.*, vol. 16, no. 4, pp. 1061–1071, 2008.
- [17] H. Ishibuchi, T. Nakashima, T. Murata, "Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems," *IEEE Trans. Syst. Man Cybern. Part B-Cybern.*, vol. 29, no. 5, pp. 601–618, 1999.
- [18] 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.*, vol. 13, no. 3, pp. 307–318, 2009. Software available online: <http://www.keel.es>