

# Bi-criteria Genetic Selection of Bagging Fuzzy Rule-based Multiclassification Systems

Krzysztof Trawiński, Arnaud Quirin, Oscar Cordon

European Centre for Soft Computing, Mieres, Spain

Email: {krzysztof.trawinski, arnaud.quirin, oscar.cordon}@softcomputing.es

**Abstract**— Previously we proposed a scheme to generate fuzzy rule-based multiclassification systems by means of bagging, mutual information-based feature selection, and a multicriteria genetic algorithm (GA) for static component classifier selection guided by the ensemble training error. In the current contribution we extend the latter component by the use of two bi-criteria fitness functions, combining the latter error measure with the selected ensemble likelihood. A study on four popular UCI datasets with different dimensionalities is conducted in order to analyze the accuracy-complexity trade-off obtained by the two GAs, the initial fuzzy ensemble and a single fuzzy classifier.

**Keywords**— Bagging, feature selection, fuzzy rule-based multiclassification systems, genetic selection of individual classifiers, multicriteria genetic algorithm.

## 1 Introduction

Multiclassification systems (MCSs) are promising data mining tools dealing with complex classification problems, especially when the number of dimensions or the size of the data are really large [1]. They usually combine decision trees [2] or neural networks [3], but also more recently fuzzy classifiers [4, 5, 6].

In a previous study [7], we described a methodology in which classical MCS design approaches such as bagging [8] and random subspace [2] are used to generate fuzzy rule-based multiclassification systems (FRBMCSs) using a basic heuristic fuzzy classification rule generation method [9], as well as a classifier selection technique based on a GA driven by a multicriteria fitness function [10]. Later, we improved our methodology in [10] using a feature selection approach based on the Battiti's method [11] and the GRASP procedure [12]. Finally, in [13], we extended the genetic selection method by considering additional error measures.

We drew the conclusion that a feature and instance selection procedure combined with a simple grid partitioning FRBS is a good approach to overcome the curse of dimensionality problem in large datasets while using FRBMCS, mainly due to the fact that these kind of classifiers are instable enough. Nevertheless, once a set of classifiers has been trained, we still need to deal with the high number of rules and the correlations between individual classifiers. This is why a selection of the classifiers is so crucial. As said before, we already proposed a GA guided by several single-criteria fitness functions, based on the training error [10], the likelihood [7], or the Out-Of-Bag error [13]. This methodology, quite novel in this topic, lead us to the generation of a compact sets of rules, while still preserving its accuracy, in a single GA run, without resorting on a Pareto-based multi-objective optimization technique. However, many experimentations suggested the choice of the

fitness function is very dependent of the problem being solved. For instance, when using the training error, the accuracies of two FRBMCSs can be similar or even null, making difficult for the GA to discriminate between them in order to improve the generalization ability. On the contrary, using the likelihood alone seems to give bad results on many datasets. This suggests the fact that a combination of some criteria could be a good idea to overcome this issue, producing better results than any criterion in isolation.

The aim of the current contribution is to propose a solution by exploring two new fitness functions based on a combination of the training error and the likelihood measures. By doing so, we will try to combine them using the two most simple ways: weighted average and lexicographic order (i.e. considering the optimization of a single criterion, and using the second in case of tie). Introducing such elaborated method we hope that it will allow the FRBMCS to deal with high dimensional data.

We aim to check if the new GA fitness functions will perform better in terms of accuracy than the previous ones for some datasets, while still being competitive for the others datasets. A preliminary study will be conducted on small and medium size datasets from the UCI machine learning repository to test the two new fitness functions in comparison to a single classifier, the original FRBMCS, and the GA-selected FRBMCSs using the said fitness functions. Several parameter settings for the global approach (e.g. different granularity levels as well as different feature selection methods) will be tested and compared regarding the accuracy and the size of the rule base obtained by a single classifier and the original FRBCS ensemble.

This paper is set up as follows. In the next section, existing GA-based methods to select MCSs are reviewed. Sec. 3 recalls our approach for designing FRBMCSs considering bagging and feature selection, while Sec. 4 describes the proposed multicriteria GA for component classifier selection. The experiments developed and their analysis are shown in Sec. 5. Finally, Sec. 6 collects some concluding remarks and future research lines.

## 2 Related work on genetic selection of MCSs

In general, the selection of a subset of classifiers is done using the *overproduce-and-choose strategy* (OCS) [14], in which a large set of classifiers is produced and then selected to extract the best performing subset. GAs are a popular technique within this strategy. In the literature, performance, complexity and diversity measures are usually considered as search criteria. Complexity measures are employed to increase the interpretability of the system whereas diversity measures are used to avoid overfitting.

Among the different genetic OCS proposals, we can remark the following ones. In [15], a hierarchical multi-objective GA (MOGA) algorithm, performing feature selection at the first level and classifier selection at the second level, is presented which outperforms classical methods for two handwritten recognition problems. The MOGA allows both performance and diversity to be considered for MCS selection. In [16] a GA is used to select from seven diversity heuristics for k-means cluster-based ensembles and the ensemble size is also encoded in the genome. In the study of Martínez-Munoz et al. [17], a GA is compared to five other techniques for ensemble selection. Even if the performance of the GA was the worst obtained, they showed that while selecting a small subset of classifiers, the generalization error was significantly decreased. In [18], the authors developed a multidimensional GA to optimize two weight-based models, in which the weights are assigned to each classifier or to each class. They applied their system to 6 different classifiers (only linear and quadratic classifiers are explored), but on only two small datasets and without comparing to the results obtained on a single classifier. Finally, our own previous studies [10, 7] also consider a multicriteria GA for the ensemble selection in an OCS fashion, with performance (training error) and complexity as criteria to guide the GA.

In general, the performance obtained with the initial MCS is outperformed by the ensemble selected by the GA, while simplifying the system. In our current contribution, we will confirm this conclusion by the study of two improved fitness functions mixing the two most used criteria: the accuracy and the complexity of the classifiers. The fitness function will directly incorporate one or two accuracy criteria (i.e., the training error and the likelihood), while the MCS complexity will be implicitly optimized by the considered coding scheme.

### 3 Bagging and feature selection-based FRBMCSs

In this section we will both detail how the individual classifiers and the FRBMCSs are designed. A normalized dataset is split into two parts, a training set and a test set. The training set is submitted to an instance selection and a feature selection procedure in order to provide individual training sets (the so-called *bags*) to train simple FRBCSs (through the method described in section 3.1). The instance selection and the feature selection procedures are described in section 3.2. After the training, we got an initial FRBMCS, which is validated using the training and the test errors (*Ensemble Training Error* and *Ensemble Test Error*), as well as a measure of complexity based on the total number of rules in the FRBCSs. This ensemble is selected using a multicriteria GA (described in the next section) guided by two accuracy-based fitness functions. The final FRBMCS is validated using different accuracy (Training Error, Test Error) and complexity measures (number of classifiers, total number of rules).

#### 3.1 Individual FRBCS composition and design method

The FRBCSs considered in the ensemble will be based on fuzzy rules  $R_j$  with a class  $C_j$  and a certainty degree  $CF_j$  in the consequent: If  $x_1$  is  $A_{j1}$  and ... and  $x_n$  is  $A_{jn}$  then Class  $C_j$  with  $CF_j$ ,  $j = 1, 2, \dots, N$ , and they will take their decisions by means of the single-winner method [9]. This fuzzy

reasoning method has been selected due to its high simplicity and interpretability. The use of other more advanced ones [19] is left for future works.

To derive the fuzzy knowledge bases, one of the heuristic methods proposed by Ishibuchi et al. in [9] is considered. The consequent class  $C_j$  and certainty degree  $CF_j$  are statistically computed from all the examples located in a specific fuzzy subspace  $D(A_j)$ .  $C_j$  is computed as the class  $h$  with maximum confidence according to the rule compatible training examples  $D(A_j) = \{x_1, \dots, x_m\}$ :  $c(A_j \Rightarrow Class\ h) = |D(A_j) \cap D(Class\ h)| / |D(A_j)|$ .  $CF_j$  is obtained as the difference between the confidence of the consequent class and the sum of the confidences of the remainder (called  $CF_j^{IV}$  in [9]).

This method is good for our aim of designing FRBCS ensembles since it is simple and quick. However, it carries two drawbacks: its low accuracy and the generation of large fuzzy rule bases. We aim to consider more advanced techniques in the future.

#### 3.2 FRBMCS design approaches

The generation of the FRBMCSs is performed by means of a bagging approach combined with a feature selection method [10]. Three different feature selection methods, random subspace and two variants of Battiti's MIFS, greedy and GRASP, are considered.

As said before, *random subspace* [2] is a method in which we select randomly a set of features from the original dataset. The greedy Battiti's MIFS method [11] is based on a forward greedy search using the Mutual Information measure [20], with regard to the class. This method selects the set  $S$  of the most informative features about the output class which cannot be predicted with the already selected features. It uses a coefficient,  $\beta$ , to set up the penalization on the information brought by the already selected features.

The MIFS-GRASP variant is an approach where the set is generated by iteratively adding features randomly chosen from a Restricted Candidate List (RCL) composed of the best  $\tau$  percent decisions according to the Battiti's quality measure. Parameter  $\tau$  is used to control the amount of randomness injected in the MIFS selection. With  $\tau = 0.5$ , we get an average amount of randomness, while still preserving the quality-based ordering of the features.

For the bagging approach, the bags are generated with the same size as the original training set, as commonly done. In every case, all the classifiers will consider the same fixed number of features.

Finally, no weights will be considered to combine the outputs of the component classifiers to take the final FRBMCS decision, but a pure voting approach will be applied: the ensemble class prediction will directly be the most voted class in the component classifiers output set.

### 4 A multicriteria GA-based MCS selection method

In our previous studies, we used a multicriteria GA, which is able to obtain a list of possible MCS designs ranked by their quality *from a single chromosome* thanks to its novel coding scheme. However, the fitness function considered was based on a single criterion, either the likelihood (L) [7], the *training*

error (TE) [10, 13], or the out-of-bag error [13]. Although the TE-based GA provided better overall results, the L-based GA outperformed it in some of the cases. That led us to the idea of combining the both measures, which moreover show complementary characteristics.

#### 4.1 Multicriteria genetic optimization

The GA searches for an optimal sequence of the classifiers, in the way that the most significant classifiers have the lowest indexes, while those redundant members, which can be safely excluded, are in the last positions. The coding scheme is thus based on an order-based representation, a permutation  $\Pi = \{j_1, j_2, \dots, j_l\}$  of the  $l$  originally generated individual classifiers. In this way, each chromosome encodes  $l$  different solutions to the problem, based on considering a “basic” MCS comprised by a single classifier, that one stored in the first gene, then another one composed of two classifiers, those in the first and the second genes, and so on.

So, the computation of the evaluation criteria for the whole ensemble is obtained in a *cumulative* way, defined as a vector containing the measured values of the first classifier; the subset formed by the first and the second; and so on. The fitness function is thus using the values of a multicriteria vector, being composed of an array of  $l$  values,  $L^i = L'_{\{j_1, j_2, \dots, j_i\}}$ , corresponding to the cumulative measure-value of the  $l$  mentioned MCS designs.

At the end of the GA, the best chromosome is that member in the population overcoming the others using the considered criterion. Then, the final design encoded in this chromosome is the MCS comprising the classifiers from the first one to the one having the best cumulative measured value (although any other design not having the optimal accuracy but, for example, showing a lowest complexity can also be directly extracted). In this way, an implicit use of a complexity criterion is also made.

To increase its convergence rate, the GA works following a steady-state approach. The initial population is composed of randomly generated permutations. In each generation, a tournament selection of size 3 is performed, and the two winners are crossed over to obtain a single offspring that directly substitutes the loser. In this study, we have considered OX crossover and the usual exchange mutation [21].

#### 4.2 The two used evaluation criteria

For the definition of the fitness functions, we use TE and L as the evaluation criteria.

The TE is computed as follows. Let  $h_1(\mathbf{x}), \dots, h_l(\mathbf{x})$  be the outputs of the component classifiers of the selected ensemble for an input value  $\mathbf{x} = (x_1, \dots, x_n)$ . For a given sample  $\{(\mathbf{x}^k, C^k)\}_{k \in \{1 \dots m\}}$ , the TE of that MCS is:

$$TE = \frac{1}{m} \cdot \#\{k \mid C^k \neq \arg \max_{j \in \{1 \dots M\}} h_j(\mathbf{x}^k)\} \quad (1)$$

Fitness evaluation using TE alone was already studied in one of our previous publications [10]. We will call it *Training Error-based Fitness Function* (TEFF).

The L is computed as follows. Let the classes  $h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_l(\mathbf{x})$  be the decisions of the component classifiers of the selected ensemble  $S$  for an input value  $\mathbf{x} =$

$(x_1, \dots, x_n)$ . We will assume that the fraction of the members of  $S$  that agree on the class of  $\mathbf{x}$  is an estimate of the conditional probability of that class:

$$P_S(C|\mathbf{x}) = \frac{1}{|S|} \cdot \#\{i \in S \mid h_i(\mathbf{x}) = C\}.$$

The L of the subset  $S$ , to be maximized, is:

$$L_S = \prod_k P_S(C^k|\mathbf{x}^k).$$

As the small values of  $L_S$  may produce numerical instabilities, we use instead a bounded log-likelihood:

$$L'_S = \sum_k \log(P_S(C^k|\mathbf{x}^k) + \epsilon),$$

where the value  $\epsilon$  foresees that case for which none of the members of the subset has found the true class of the pattern. In [7], we endowed the fitness function with the L, as it allows us to discern differences between ensembles with the same TE (specifically, between those with null error!). A learning process using only the TEFF will automatically end up with the learning, while L will go on improving the estimations of the probability distributions for each class, thus reducing the chances of overfitting the training data.

#### 4.3 The two new bi-criteria fitness functions

In this contribution we propose two approaches for the fitness function combining the L with the TE measure, the *Lexicographical Order-based Fitness Function* (LOFF) and the *Weighted Combination Fitness Function* (WCFF).

Notice that, working in this way, we are introducing a second multi-criteria optimization level in our algorithm. On the one hand, a multi-criteria optimization is made by means of the considered coding scheme and the cumulative evaluation of the possible MCS designs (see Sect. 4.1). On the other hand, a higher level is introduced when evaluating the latter possible designs by means of a bi-criteria fitness function.

In the first one, the LOFF, we use the lexicographical order to deal with the multicriteria optimization. When comparing two chromosomes, one is better than the other if it takes a better (lower) minimum value of the TE. In case of tie, the L measure is considered. The ordering scheme gives priority to TE, as it provided better results in our previous study, while taking the L only in the last resort in the case of the frequent ties encountered by the system.

In the second approach, the WCFF, we propose objective function scalarization by a weighted combination of both measures:

$$WC = factor_0 * \alpha * TE + (1 - \alpha) * L \quad (2)$$

where  $\alpha$  is a weight in  $[0,1]$  and  $factor_0 = L_0/TE_0$  is a first evaluation-based normalization using  $L_0$  and  $TE_0$ , the L and the TE from the initial FRBMCS. The fitness function is to be minimized.

## 5 Experiments and analysis of results

In this section, we discuss the performance obtained by a single FRBCS, the initial FRBMCS, and three different GA-selected FRBMCSs including our two new fitness functions on four chosen datasets.



### 5.1 Experimental setup

To evaluate the performance of the generated FRBMCSs, we have selected four datasets from the UCI machine learning repository (see Table 1). In order to compare the accuracy of the considered classifiers, we used Dietterich's  $5 \times 2$ -fold cross-validation ( $5 \times 2$ -cv), which is considered to be superior to paired  $k$ -fold cross validation in classification problems [22].

Table 1: Data sets considered

Data set	#attr.	#examples	#classes
Pima	8	768	2
Glass	9	214	7
Vehicle	18	846	4
Sonar	60	208	2

Three different granularities, 3, 5 and 7, are tested for the single FRBCS derivation method, for feature sets of size 5 selected by means of three approaches: the greedy Battiti's MIFS filter feature selection method [11], the Battiti's method with GRASP (with  $\tau$  equal to 0.5, see section 3.2), and random subspace [2]. Battiti's method has been run by considering a discretization of the real-valued attribute domains in ten parts and setting the  $\beta$  coefficient to 0.1.

The FRBMCSs generated are initially comprised by 50 classifiers. The GA for the component classifier selection works with a population of 50 individuals and runs during 50 generations. The mutation probability considered is 0.05. The weights of WCFF were set to 0.8 for TE and 0.2 for L as our aim was to allow a small influence of the L in the cases in which the TE gives similar values. The other values for the weights will not improve the results significantly.

All the experiments have been run in a cluster at the University of Granada on Intel quadri-core Pentium 2.4 GHz nodes with 2 GBytes of memory, under the Linux operating system.

### 5.2 Comparison of the three fitness functions

The statistics ( $5 \times 2$ -cv error, number of rules, and run time required for each run, expressed in seconds) for the genetically selected FRBCS ensembles using LOFF, WCFF and TEFF are collected in Tables 2, 3 and 4 respectively. There are three subtables for each of the feature selection method considered. The best results for a given feature selection method are shown in bold and the best values overall are outlined.

Comparing the three fitness functions, we can see how the WCFF approach is able to outperform the TEFF and LOFF considering the individual test error 9 times (+2 draw). The best individual improvement was observed on the sonar dataset (-8% regarding LOFF, -4% regarding TEFF) with greedy and 5 labels. We observed that in 4 out of 9 cases, WCFF outperforms other approaches on the sonar and pima datasets. The best overall result was obtained on the pima dataset with GRASP and 5 labels (draw with LOFF).

The FRBMCSs based on LOFF are better than TEFF and WCFF in 12 of the 36 cases (+2 draw). The best individual improvement was observed on the sonar dataset with random subspace and 5 labels (-9% regarding WCFF, -12% regarding TEFF). The best overall result was obtained on the pima dataset with greedy and 5 labels (draw with WCFF) and on

the glass dataset with GRASP and 5 labels. We can observe that in 5 out of 9 cases, LOFF outperforms the other genetic approaches on the glass dataset.

Comparing the two new fitness functions, the LOFF provides better results than WCFF considering individual test error in 21 cases (+2 draw). We can also observe that LOFF outperforms the other approach on the glass dataset in 7 out of 9 cases (+1 draw) giving a good performance for the vehicle (7 out of 9 cases) datasets. However, the WCFF is better in 6 out of 9 cases on the pima dataset and in 5 out of 9 cases (+1 draw) on the sonar dataset. In general, LOFF is better than WCFF since it gives a lower influence to L, whereas the weighted combination is better on the sonar dataset due to the many ties with the TE on this dataset, making the use of the L more likely.

The TEFF-based FRBMCSs outperforms the LOFF and the WCFF considering individual test error for 13 of the 36 times. The best overall result was obtained on the sonar dataset with random subspace and 3 labels, and on the vehicle with random subspace and 7 labels. We may conclude that the LOFF and WTEL are competitive with TEFF. The LOFF got best results 12 times (+2 draw), WTEL 9 (+2 draw) and TEFF 13 times. However, in the direct comparison, the use of LOFF improves the single TEFF performance in 18 out of 36 cases (+3 draw) and the WTEL improves the single TEFF performance in 19 out of 36 cases, which indicates that L does not produce overfitting. Thus, the use of L as a secondary criterion is only useful in some of the cases.

On 36 cases, the number of classifiers is lower in 12 cases (+1 draw) with the LOFF, in 9 cases with the WCFF and in 15 cases (+1 draw) with the TEFF. The two new fitness functions generate a higher number of classifier, since they are more conservative due to the use of L. Such fitness functions could be viewed as a proper way to improve performance of the datasets having the larger size.

### 5.3 Genetically selected FRBMCSs vs. single FRBCS/original FRBMCSs

The results of the single FRBCSs are presented in Table 5 while those of the original FRBMCSs are presented in Table 6. In all the 36 cases, the generated FRBMCSs improve the performance of the single FRBCS.

Although the main goal of the genetic selection is to reduce the complexity of the generated FRBMCS, the accuracy results obtained from that process are also improved in most of the cases, showing the potential of the approach. In only 10 of the 36 cases (+1 draw) the original FRBMCS outperforms the best genetically designed one in terms of accuracy. Comparing the best overall TE values of genetically selected FRBMCSs with those of the original FRBMCSs, the GA improves the results on the sonar dataset (-2% regarding TEFF) and glass (-10% regarding LOFF), showing a slight increase for the other problems (+2% for pima regarding LOFF/WTEL, +2% for vehicle regarding TEFF).

### 5.4 Statistical significance of the results

Table 7 shows the results of the statistical tests performed to check if the performance of the initial FRBMCSs and the performance of the GA selected FRBMCSs outperform significantly the performance of the single classifier. The Wilcoxon signed-rank test [23] has been used for this purpose. The best

# IFSA-EUSFLAT 2009

Table 2: Results for the FRBCS ensembles selected by the GA using the LOFF

		Bagging + Greedy				Bagging + Random Subspace				Bagging + GRASP $\tau = 0.50$			
		Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar
3 labels 5 #attr.	5×2-cv	0.257	<b>0.363</b>	0.468	<b>0.246</b>	0.261	<b>0.375</b>	0.423	<b>0.219</b>	0.253	0.367	0.440	<b>0.227</b>
	#classifiers	3.8	10.1	9.5	8.5	4.0	13.0	14.2	19.9	4.2	9.1	11.7	12.9
	#rules	659.7	1275.9	1298.6	1287.2	668.4	1450.0	2336.3	3363.0	742.4	1162.3	1795.1	2110.9
	avg. #rules time	176.6 599.57	126.5 127.00	138.3 665.84	151.0 164.46	167.6 436.40	112.5 111.97	165.3 482.74	169.9 117.36	175.2 534.17	125.9 134.80	152.0 591.25	163.6 144.59
5 labels 5 #attr.	5×2-cv	<b>0.238</b>	0.374	0.391	0.259	<b>0.257</b>	0.387	0.374	0.223	<b>0.240</b>	<b>0.358</b>	0.399	0.234
	#classifiers	12.9	8.5	14.0	10.1	12.9	14.2	14.2	14.1	10.9	16.0	13.4	13.0
	#rules	7491.1	2211.3	6711.9	5696.9	7160.0	3439.0	10374.3	9273.3	6535.1	4190.4	7610.7	8291.5
	avg. #rules time	582.0 434.14	268.1 121.72	482.1 487.14	569.4 117.84	592.2 599.22	247.5 167.95	740.3 665.79	680.7 162.04	599.5 370.50	271.7 102.98	582.1 405.34	644.4 100.62
7 labels 5 #attr.	5×2-cv	0.258	0.387	<b>0.375</b>	0.258	0.270	0.393	<b>0.347</b>	0.275	0.258	0.375	<b>0.355</b>	0.250
	#classifiers	14.4	9.5	14.0	7.3	15.8	13.4	17.8	6.2	13.3	9.1	15.5	8.5
	#rules	18851.1	3786.4	16266.7	7609.4	19619.0	5169.0	29596.5	7690.1	17892.8	3738.7	20978.9	10210.4
	avg. #rules time	1208.1 369.73	400.2 103.94	1146.7 405.34	1056.8 100.18	1240.5 366.53	392.6 103.25	1686.0 407.70	1234.8 99.94	1338.9 498.07	409.5 140.94	1386.4 582.16	1214.6 135.59

Table 3: Results for the FRBCS ensembles selected by the GA using the WCFF

		Bagging + Greedy				Bagging + Random Subspace				Bagging + GRASP $\tau = 0.50$			
		Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar
3 labels 5 #attr.	5×2-cv	0.254	<b>0.364</b>	0.488	0.256	0.258	0.384	0.430	<b>0.224</b>	0.255	<b>0.370</b>	0.459	<b>0.226</b>
	#classifiers	6.2	8.8	15.6	6.8	6.3	10.5	17.7	22.2	7.5	13.9	10.3	10.7
	#rules	1081.6	1139.8	2209.2	1030.0	1020.8	1190.4	2944.8	3683.5	1331.4	1722.5	1587.2	1677.3
	avg. #rules time	174.3 401.63	125.4 111.35	140.2 441.11	153.2 108.23	161.2 382.65	113.1 105.19	166.6 419.06	171.0 102.40	172.4 392.34	126.3 108.28	153.3 428.73	167.6 105.53
5 labels 5 #attr.	5×2-cv	<b>0.243</b>	0.380	0.395	<b>0.238</b>	<b>0.256</b>	<b>0.382</b>	0.372	0.244	<b>0.238</b>	0.386	0.401	0.234
	#classifiers	17.4	12.9	14.7	14.9	16.3	9.2	16.0	19.1	14.6	13.0	15.4	22.4
	#rules	10118.3	3438.4	7258.3	8202.4	9104.2	2347.5	11739.3	12502.4	8686.7	3457.0	8947.0	13497.0
	avg. #rules time	589.8 383.37	264.5 106.09	486.4 420.65	566.8 102.93	550.0 401.48	260.3 111.36	737.4 441.42	674.3 108.02	593.8 371.67	270.3 102.42	593.6 411.51	622.2 100.52
7 labels 5 #attr.	5×2-cv	0.250	0.402	<b>0.368</b>	0.256	0.272	0.393	<b>0.353</b>	0.275	0.254	0.383	<b>0.363</b>	0.243
	#classifiers	13.8	13.4	12.6	8.0	16.2	15.1	10.3	6.2	14.8	11.7	12.1	9.7
	#rules	17992.3	5231.7	14853.7	8248.2	20343.7	5923.8	18443.5	7690.1	19726.7	4728.6	17199.3	11271.4
	avg. #rules time	1323.8 372.49	399.7 102.26	1164.6 407.72	1053.8 100.36	1258.5 368.65	411.1 103.92	1814.7 408.42	1234.8 99.82	1343.0 386.31	402.2 107.54	1435.8 425.35	1173.8 104.44

Table 4: Results for the FRBCS ensembles selected by the GA using the TEFF

		Bagging+Greedy				Bagging+GRASP $\tau = 0.50$				Bagging + Random Subspace			
		Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar
3 labels 5 #attr.	5×2-cv	0.257	<b>0.360</b>	0.461	<b>0.235</b>	0.254	0.372	0.449	<b>0.237</b>	<b>0.256</b>	<b>0.381</b>	0.428	<b>0.216</b>
	#classifiers	4.1	7.3	10.3	12.3	14.4	10.2	12.9	13.9	4.2	13.7	13.4	20.1
	#rules	696.5	904.3	1431.0	1842.1	763.0	1317.9	1991.6	2252.6	703.4	1546.0	2239.5	3376.7
	avg. #rules time	171.5 94.06	125.4 26.35	138.3 103.26	148.3 25.32	174.3 93.37	126.0 26.49	155.9 102.09	161.7 25.18	168.1 92.77	113.1 26.39	168.9 103.24	168.3 25.08
5 labels 5 #attr.	5×2-cv	<b>0.242</b>	0.383	0.392	0.247	<b>0.239</b>	<b>0.363</b>	0.399	0.252	0.263	0.392	0.378	0.249
	#classifiers	11.5	15.9	15.5	10.4	10.9	14.7	12.0	7.8	11.9	13.7	13.0	9.4
	#rules	6744.9	4233.1	7338.4	5757.7	6497.4	3986.7	7227.3	4893.9	6680.0	3312.2	9455.9	6208.8
	avg. #rules time	592.8 93.48	268.7 26.10	481.9 103.48	567.0 25.17	593.5 92.58	282.0 26.16	611.3 103.75	630.0 24.86	555.8 91.47	245.0 26.18	734.3 104.81	668.8 24.83
7 labels 5 #attr.	5×2-cv	0.258	0.393	<b>0.374</b>	0.258	0.256	0.395	<b>0.356</b>	0.257	0.265	0.393	<b>0.337</b>	0.267
	#classifiers	12.7	8.9	14.6	6.3	16.4	10.3	13.2	6.7	17.0	15.5	17.5	6.4
	#rules	16614.3	3524.3	16102.3	6427.0	21836.6	4140.6	18296.2	7767.8	21289.5	5980.6	28854.2	7655.2
	avg. #rules time	1313.9 92.87	404.5 26.50	1115.7 102.90	1040.9 24.85	1346.2 92.49	401.9 26.18	1386.5 102.93	1148.7 25.31	1248.4 92.31	386.2 26.08	1680.2 103.52	1203.7 25.19

Table 5: Results for the single FRBCSs with feature selection

		Greedy				GRASP $\tau = 0.50$				Random Subspace			
		Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar
3 labels 5 #attr.	5×2-cv	0.266	0.446	0.549	<b>0.261</b>	0.267	0.447	0.546	0.316	0.265	0.457	0.512	<b>0.319</b>
	#rules	178.50	135.30	136.40	146.60	179.50	137.00	135.80	169.00	161.80	109.50	154.50	174.50
	avg. #rules time	0.08 0.08	0.04 0.04	0.12 0.12	0.08 0.08	0.09 0.09	0.04 0.04	0.12 0.12	0.09 0.09	0.07 0.07	0.03 0.03	0.12 0.12	0.08 0.08
	5 labels 5 #attr.	5×2-cv	<b>0.246</b>	<b>0.376</b>	0.430	0.287	<b>0.246</b>	<b>0.375</b>	0.425	<b>0.314</b>	<b>0.262</b>	0.435	0.460
#rules	682.70	291.00	437.60	615.20	682.70	293.50	418.90	752.70	604.20	259.60	587.80	773.60	
avg. #rules time	0.42 0.42	0.25 0.25	0.65 0.65	0.16 0.16	0.39 0.39	0.26 0.26	0.63 0.63	0.17 0.17	0.36 0.36	0.24 0.24	0.67 0.67	0.17 0.17	
7 labels 5 #attr.	5×2-cv	0.262	0.414	<b>0.402</b>	0.291	0.266	0.423	<b>0.399</b>	0.317	0.276	<b>0.418</b>	<b>0.415</b>	0.340
	#rules	1600	431.20	1021	1218	1599	437.20	907.50	1470	1432	410.90	1266	1536
	avg. #rules time	1.75 1.75	1.32 1.32	3.27 3.27	0.52 0.52	1.71 1.71	1.34 1.34	3.25 3.25	0.55 0.55	1.66 1.66	1.32 1.32	3.37 3.37	0.63 0.63

Table 6: Results for the FRBCS ensembles

		Bagging+Greedy				Bagging+GRASP $\tau = 0.50$				Bagging + Random Subspace			
		Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar
3 labels 5 #attr.	5×2-cv	0.261	0.463	0.525	0.255	0.262	0.464	0.494	0.246	0.299	0.450	0.453	0.250
	#rules	857.8	620.8	684.3	728.2	860.9	628.9	736.2	795.1	793.6	567.1	800.8	817.4
	avg. #rules time	171.55 3.43	124.16 1.51	136.87 4.87	145.65 2.52	172.18 3.45	125.77 1.53	147.24 4.91	159.03 2.57	158.71 3.34	113.42 1.49	160.16 5.06	163.47 2.58
	5 labels 5 #attr.	5×2-cv	<b>0.235</b>	<b>0.396</b>	0.400	<b>0.240</b>	<b>0.234</b>	<b>0.405</b>	0.399	<b>0.220</b>	<b>0.260</b>	0.430	0.378
#rules	2940.5	1287.7	2217.7	2676.9	2974.8	1330.2	2557.8	3006.8	2719.9	1199.8	3079.9	3182.4	
avg. #rules time	588.11 17.93	257.54 12.11	443.55 31.21	535.37 6.66	594.95 18.05	266.04 12.23	511.56 32.79	601.36 6.96	543.97 17.64	239.96 11.94	615.97 33.91	636.47 7.13	
7 labels 5 #attr.	5×2-cv	0.243	0.430	<b>0.375</b>	0.262	0.247	0.425	<b>0.353</b>	0.242	0.263	<b>0.402</b>	<b>0.330</b>	0.241
	#rules	6489.1	1863.3	4847.9	4958.7	6580.2	1927.2	5472.1	5468.4	5982.4	1799.9	6793.6	5729.8
	avg. #rules time	1298 84.70	372.66 67.36	969.58 166.51	991.74 24.72	1316 85.27	385.45 68.27	1094 170.48	1094 25.49	1196 82.12	359.98 66.06	1359 174.24	1146 25.57

Table 7: Statistical test for the comparison of the single FRBCS and the different FRBMCSs methodology. For each dataset, the best result is marked (\*\*) and the others are compared to it

		Best single classifier (app./labels)	Best ensemble (app./labels)	Best ens. selected (app./labels/fitness)	Best ens. TEFF (app./labels)
Pima	Approach	GRASP/5	GRASP/5	Greedy/5/LOFF	GRASP/5
	$\mu \pm \sigma$	0.246 ± 0.00991	0.234 ± 0.019	0.238 ± 0.0167	0.240 ± 0.0159
	Symbol	=	*	=	=
Glass	Approach	GRASP/5	Greedy/5	Greedy/3/LOFF	GRASP/3
	$\mu \pm \sigma$	0.375 ± 0.0526	0.396 ± 0.0568	0.358 ± 0.0382	0.360 ± 0.0507
	Symbol	=	=	*	=
Vehicle	Approach	GRASP/7	Random/7	Random/7/LOFF	GRASP/7
	$\mu \pm \sigma$	0.399 ± 0.0262	0.330 ± 0.0179	0.347 ± 0.0196	0.337 ± 0.0168
	Symbol	+	*	*	=
Sonar	Approach	Greedy/3	GRASP/5	Random/5/WCFF	GRASP/3
	$\mu \pm \sigma$	0.261 ± 0.0463	0.220 ± 0.0445	0.219 ± 0.0264	0.216 ± 0.0265
	Symbol	+	=	=	*

result for each dataset is marked with a star '\*'.

The best results (in average) are always obtained by the initial or the selected ensembles. Even if they are only significant for two datasets, they correspond to those with the largest dimension. Notice that, on sonar, the GA outperforms significantly the single classifier. The initial ensemble outperforms the best genetically selected FRBMCSs in one of the four cases (vehicle), whereas TEFF achieves the best result for the sonar dataset and WCFF for the sonar.

In the direct comparison, TEFF is the best choice two times for the vehicle and sonar dataset. The new fitness functions outperforms one of the four cases for pima with LOFF and glass dataset with WCFF.

Thus, combining bagging and the GA selection process to design FRBMCSs performs better for high dimensional problems with a large number of attributes, producing a smaller rule base while reducing the test errors in some cases, which was our original goal. When combining these two techniques with an advanced feature selection process we also get an improvement of the accuracy for datasets with higher dimensions (glass, vehicle and sonar, see Table 7).

## 6 Conclusions and future works

In this study, we extended our previously developed methodology in which a bagging approach together with a feature selection technique are used to train FRBMCSs, at a later stage selected by a multicriteria GA. Two new fitness functions were tested, the LOFF and the WCFF, based on one or two accuracy criteria (i.e., the training error and the likelihood). The generated selected FRBCS ensembles are performing correctly on classification problems with a significant number of features. By using above-mentioned techniques, we would like to obtain FRCMCS dealing with high dimensional data.

One of the next steps we will consider in the future line is the design of a generic framework to define the multicriteria fitness function. At least two different information levels will be studied: the chromosome and the objective level. Furthermore, we would like to extend this study on larger data sets (more than 1,000 examples), to study the influence of other parameters (the GA parameters, the weighting coefficient in the WCFF, etc.), and to design more advanced genetic MCS selection techniques (e.g. the use of Pareto-based algorithms). Analysis of the different fuzzy rule generation techniques and introduction a diversity criterion in the algorithm are another important points for future research.

## References

- [1] L. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.
- [2] T. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
- [3] D. Optiz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.
- [4] J. Canul-Reich, L. Shoemaker, and L. O. Hall. Ensembles of fuzzy classifiers. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6, London, 2007.
- [5] W. Pedrycz and K.C. Kwak. Boosting of granular models. *Fuzzy Sets and Systems*, 157(22):2934–2953, 2006.
- [6] S.J. Verzi, G.L. Heileman, and M. Georgiopoulos. Boosted ARTMAP: Modifications to fuzzy ARTMAP motivated by boosting theory. *Neural Networks*, 19(4):446–468, 2006.
- [7] O. Cordón, A. Quirin, and L. Sánchez. A first study on bagging fuzzy rule-based classification systems with multicriteria genetic selection of the component classifiers. In *IEEE International Workshop on Genetic and Evolving Fuzzy Systems (GEFS)*, pages 11–16, Germany, 2008.
- [8] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
- [9] H. Ishibuchi, T. Nakashima, and M. Nii. *Classification and Modeling With Linguistic Information Granules*. Springer, 2005.
- [10] O. Cordón, A. Quirin, and L. Sánchez. On the use of bagging, mutual information-based feature selection and multicriteria genetic algorithms to design fuzzy rule-based classification ensembles. In *Int. Conf. on Hybrid Intelligent Systems*, pages 549–554, Barcelona, 2008.
- [11] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994.
- [12] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *J. of Global Optimization*, 6:109–133, 1995.
- [13] O. Cordón and A. Quirin. Comparing two genetic overproduce-and-choose strategies for fuzzy rule-based multiclassification systems generated by bagging and mutual information-based feature selection. Technical Report #AFE 2008-05, European Centre for Soft Computing, Mieres, 2008.
- [14] D. Partridge and W.B. Yates. Engineering multiversion neural-net systems. *Neural Computation*, 8(4):869–893, 1996.
- [15] L.S. Oliveira, M. Morita, R. Sabourin, and F. Bortolozzi. Multi-objective genetic algorithms to create ensemble of classifiers. *Lecture Notes in Computer Science*, 3410:592–606, 2005.
- [16] S.T. Hadjitodorov and L.I. Kuncheva. Selecting diversifying heuristics for cluster ensembles. *Lecture Notes in Computer Science*, 4472:200–209, 2007.
- [17] G. Martínez-Munoz, D. Hernández-Lobato, and A. Suárez. Selection of decision stumps in bagging ensembles. *Lecture Notes in Computer Science*, 4668:319–328, 2007.
- [18] B. Gabrys and D. Ruta. Genetic algorithms in classifier fusion. *Applied Soft Computing*, 6(4):337–347, 2006.
- [19] O. Cordón, M.J. del Jesus, and F. Herrera. A proposal on reasoning methods in fuzzy rule-based classification systems. *International J. of Approximate Reasoning*, 20:21–45, 1999.
- [20] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.
- [21] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, 1996.
- [22] T.G. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.
- [23] E. Lehmann. *Nonparametric Statistical Methods Based on Ranks*. McGraw-Hill, New-York, 1975.