

# On the use of bagging, mutual information-based feature selection and multicriteria genetic algorithms to design fuzzy rule-based classification ensembles

Oscar Cordon, Arnaud Quirin and Luciano Sánchez<sup>‡</sup>

## Abstract

*In this contribution we explore the combination of bagging with random subspace and two variants of Battiti's mutual information feature selection methods to design fuzzy rule-based classification system ensembles. Besides, we consider a multicriteria genetic algorithm guided by the training error to select the component classifiers, in order to look for appropriate accuracy-complexity trade-offs in the final multiclassifier.*

## 1 Introduction

Classifier ensembles or multiclassifiers have been shown as very promising tools to improve the performance of single classifiers when dealing with complex, high dimensional classification problems in the last few years [1]. This research topic has become especially active in the classical machine learning area, considering decision trees or neural networks to generate the component classifiers, but also some work has been done using different kinds of fuzzy classifiers (see [2] for a review). In that previous study, we described fuzzy rule-based classification systems (FRBCS) ensembles from classical approaches such as bagging [3] and random subspace [4] with a basic, heuristic fuzzy classification rule generation method [5]. We applied a multicriteria genetic algorithm (GA) [15] guided by a likelihood measure for component classifier selection, allowing us to both improve their accuracy and to make them fit with high dimensional classification problems.

\*This work has been supported by the Spanish Ministry of Education and Science, under grants TIN2005-08036-C05-05 and TIN2006-00829.

<sup>†</sup>O. Cordon and A. Quirin are with European Centre for Soft Computing, Edificio Científico-Tecnológico, planta 3. Gonzalo Gutiérrez Quirós, s/n, 33600 - Mieres (Asturias), SPAIN {oscar.cordon, arnaud.quirin}@softcomputing.es

<sup>‡</sup>L. Sánchez is with Computer Science Department. University of Oviedo, Campus de Viesques, 33203 - Gijón (Asturias), SPAIN luciano@uniovi.es

The aim of the current contribution is to take a step ahead on those first developments by paying more attention to the feature selection process. To do so, we will consider a more advanced approach than random selection based on the use of mutual information measures. The classical Battiti's mutual information feature selection (MIFS) method [6], a greedy heuristic, and its extension to a greedy randomized adaptive search procedure (GRASP) [7] will be both considered for the generation of the component FRBCSs and compared to simple random subspace, all of them together with the bagging resampling.

The combination of the latter generation approach and a multicriteria GA for component classifier selection guided by the training error will allow us to generate FRBCS ensembles with different accuracy-complexity trade-offs in a single run. The new proposals will be tested on four popular data sets from the UCI machine learning repository with different characteristics, providing results both with and without the genetic selection stage.

The resulting techniques will thus belong to the genetic fuzzy systems family, one of the most successful approaches to hybridize fuzzy systems with learning and adaptation methods in the last fifteen years [8], and will be quite novel in the fuzzy systems area as no previous proposal considering neither bagging nor joint bagging and feature selection has been done in the literature up to our knowledge.

This paper is set up as follows. In the next section, popular classifier ensemble design approaches are reviewed. Sec. 3 recalls our approach for designing FRBCS ensembles considering bagging and feature selection, while Sec. 4 describes the proposed GA for component classifiers selection. The experiments developed and their analysis are shown in Sec. 5. Finally, Sec. 6 collects some concluding remarks and future research.

## 2 Background and related work

An ensemble of classifiers (also called a multiclassifier) is the result of the combination of the outputs of a group of

individually trained classifiers in order to get a system that is usually more accurate than any of its single components [1]. There are different ways to design these ensembles. On the one hand, there is a classical group of approaches considering *data resampling* to obtain different training sets to derive each individual classifier. In *bagging* [3], the individual classifiers are independently learnt from resampled training sets (“bags”), which are randomly selected with replacement from the original training data set. *Boosting* methods [9] sequentially generate the individual classifiers by selecting the training set for each of them based on the performance of the previous classifier(s) in the series. Opposed to bagging, the resampling process gives a higher probability of selection to the incorrectly predicted examples by the previous classifiers.

On the other hand, a second group can be found comprised by a set of approaches using some alternative ways to induce the component classifier diversity different from resampling. Feature selection plays a key role in many of them where each classifier is derived by considering a different subset of the original features. *Random subspace* [4], where each feature subset is randomly generated, is one of the most representative methods of this kind. Other generic approaches considering more advanced feature selection strategies are to be found in [10].

Finally, there are some advanced proposals that can be considered as *combinations of the two groups*. The most extended one could be *random forests* [11], where the individual classifiers are decision trees learnt from a resampled “bag” of examples, a subset of random variables is selected at each construction step, and the best split for those selected variables is chosen for that node.

The interested reader is referred to [12, 13] for two reviews for the case of decision tree ensembles (both) and neural networks (the latter), including exhaustive experimental studies. For a short review on the existing fuzzy classifier ensemble generation approaches based on bagging, boosting and feature selection, the interested reader is referred to our previous study [2]. Up to our knowledge, apart from our proposals in that same contribution, no previous work has been done on bagging FRBCSs.

### 3 Bagging and feature selection-based FRBCS ensembles

In this section we will both detail how the individual classifiers and the ensembles are designed.

#### 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 [5]. To derive the fuzzy knowledge bases, one of the heuristic methods proposed by Ishibuchi et al. in [5] is considered:  $C_j$  is computed as the class  $h$  with maximum confidence according to the rule compatible training examples  $D(A_j) = \{x_1, \dots, x_m\}$ :

$$c(A_j \Rightarrow \text{Class } h) = \frac{|D(A_j) \cap D(\text{Class } h)|}{|D(A_j)|} = \frac{\sum_{p \in \text{Class } h} \mu_{A_j}(x_p)}{\sum_{p=1}^m \mu_{A_j}(x_p)}; \quad h = 1, 2, \dots, M;$$

$CF_j$  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 [5]):

$$CF_j = c(A_j \Rightarrow \text{Class } C_j) - \sum_{h=1; h \neq C_j}^m c(A_j \Rightarrow \text{Class } h).$$

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 FRBCS ensemble design approaches

In this contribution we are applying a bagging approach in order to generate FRBCS ensembles. Three different feature selection methods, random subspace and two variants of Battiti’s MIFS, greedy and GRASP, are considered.

As said before, *random subspace* [4] is a method in which we select randomly a set of features from the original data. The greedy Battiti’s MIFS method [6] is based on a forward greedy search using the Mutual Information measure [14], 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. The Mutual Information  $I(C, F)$  for a given feature  $F$  is defined as:

$$I(C, F) = \sum_{c, f} P(c, f) \log \frac{P(c, f)}{P(c)P(f)} \quad (1)$$

where  $P(c)$ ,  $P(f)$  and  $P(f)$  are respectively the values of the density function for the class and the feature variables, and the joint probability density. In the MIFS method, we select as a first feature  $f$ , the one that maximizes  $I(C, f)$ , and then the features  $f$  that maximize  $Q(f) = I(C, f) - \beta \sum_{s \in S} I(f, s)$ , until  $S$  reaches the desired size.  $\beta$  is a coefficient 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  $Q$  measure. Parameter  $\tau$  is used to control the amount of randomness injected in the MIFS selection. With  $\tau = 0$ , we get the original MIFS method, while with  $\tau = 1$ , we get the random subspace method.

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.

In order to keep the interpretability of the generated classifier ensembles as high as possible, two decisions has been made. First, no weights will be considered to combine the outputs of the component classifiers to take the final multiclassifier 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. Besides, a multicriteria GA for classifier selection will be applied to design the optimal ensemble from the individual classifiers derived by each of the former approaches.

#### 4 A multicriteria genetic-based classifier ensemble selection method

We propose to use a multicriteria GA in order to be able not only to obtain a single solution, i.e., a classifier ensemble composition, but a list of possible ensemble designs, ranked by their quality (cumulative training error), *from a single chromosome*.

This training error is computed as follows. Let  $h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_l(\mathbf{x})$  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 training error of that ensemble is:

$$\frac{1}{m} \cdot \#\{k \mid C^k \neq \arg \max_{j \in \{1 \dots M\}} h_j(\mathbf{x}^k)\}, \quad (2)$$

Notice that, in our previous study [2], we used the same GA approach, but considering the likelihood instead of the training error as the error measure, as it seems to be more appropriate when basic feature selection methods are used.

The GA looks for an optimal ordering of the component classifiers, so that the most relevant classifiers have the lowest indices and those redundant members that can be safely discarded are in the last places. 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” ensemble comprised by a single classifier, that one stored

in the first gene; another one composed of two classifiers, those in the first and the second genes, and so forth.

The degree to which a permutation fulfills this goal is measured by means of the *cumulative training error* of the ensemble, defined as the vector containing the training error values of the first classifier; the subset formed by the first and the second; and so on. The fitness function is thus multicriteria, being composed of an array of  $l$  values,  $L^i = L'_{\{j_1, j_2, \dots, j_i\}}$ , corresponding to the cumulative training error of the  $l$  mentioned ensemble designs. The best chromosome is that member in the population with the lowest minimum cumulative training error. Then, the final design is the ensemble comprising the classifiers from the first one to the one having the minimum cumulative training error value (although any other design not having the optimal training error but, for example, showing a lowest complexity can also be directly extracted).

Instead of using a Pareto-based approach [15], a lexicographical order is considered to deal with the multicriteria optimization, since we think it better matches our scenario. When comparing two chromosomes, one is better than the other if it takes a better (lower) minimum value of the cumulative training error. In case of tie, the first positions of the fitness arrays are compared. If both first positions are of equal value, the second position is compared, and so on.

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.

## 5 Experiments and analysis of results

In this section, we discuss the performance obtained by a single FRBCS, an FRBCS ensemble and a GA-selected FRBCS ensemble on four selected data sets.

### 5.1 Experimental setup

To evaluate the performance of the FRBCS ensembles generated, we have selected four data sets 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) [16]. Three different granularities, 3, 5 and 7, are tested for the single FRBCS derivation method, as well as it is run by considering two different feature sets of size 3 and 5 selected by means of three approaches: the greedy Battiti’s MIFS filter feature selection method, the Battiti’s method with GRASP (with  $\tau$  equal to 0.25 and 0.5), and random subspace. Battiti’s method has been run by considering a discretization of

the real-valued attribute domains in ten parts and setting the  $\beta$  parameter to 0.1.

**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

The FRBCS ensembles 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.

All the experiments have been run in an Intel dual-core Pentium 2.8 GHz computer with 2 GBytes of memory, under the Linux operating system.

## 5.2 Single FRBCS vs. bagging + feature selection FRBCS ensembles

The statistics (5 $\times$ 2-cv error, number of rules and run time required for each run, expressed in seconds) for the single FRBCSs are collected in Table 2. The best results for a given feature selection method are shown in bold, and the best overall are outlined. Those results for the FRBCS ensembles of 50 classifiers generated from the four different selection approaches considered are shown in Table 3. There are four blocks in each table for each feature selection method considered: Battiti’s method (greedy), Battiti’s method combined with GRASP with 25% of randomness (GRASP 0.25), GRASP-based Battiti’s method with 50% of randomness (GRASP 0.50), and the random subspace method.

As in our previous study [2], the best results were obtained using 5 labels for the smaller problems (pima and glass), and 7 labels for the largest ones (vehicle and sonar). We have also demonstrated that a feature selection is always needed to obtain the best results for the largest problems. For example, for sonar, the best results without feature selection is 0.250 using 3 labels, that is outperformed by the GRASP 0.50 selection (0.209 using 5 labels and 0.200 using 7 labels).

With a single FRBCS, the best results were obtained with the greedy and the GRASP 0.25 approaches for the smaller datasets (pima and glass), but with GRASP 0.50 for the largest dataset (sonar), thus confirming the fact that randomness in the feature selection process is useful when combined with FRBCSs.

The bagging+feature selection approach is able to give better or almost the same accuracy than a single FRBCS, but with a smaller number of rules, for the largest dataset. The

single FRBCS gets a test error of 0.200 using the GRASP 0.50 approach with 7 labels and 3 attributes (best result for sonar) using 7662 rules, while the bagging+feature selection approach gets a slightly worst test error of 0.202 using the same approach, but with only 6895 rules. This confirms our assumption that bagging is able to give a good accuracy-complexity trade-off, when combined with a highly random feature selection approach.

Finally, over all the different feature selection approaches, for the sonar dataset, the bagging+feature selection approach allowed a decrease of 2% of the test error, while reducing by 10% the average size of the individual classifiers. The best example is produced with the greedy approach, using 5 labels and 5 attributes, in which the bagging allowed us to get a decrease of 16% of the test error, while reducing the size of the rule base by a 13%.

## 5.3 FRBCS ensemble genetic selection

The values for the genetically selected FRBCS ensembles are collected in Table 4. The first conclusion we can draw is that the GA was able to reduce the best test error for the first two problems, in comparison with a single classifier, and even the best test error obtained with the FRBCS ensembles for the glass problem. For the two last problems, the best test error slightly increases in comparison to the best FRBCS ensembles (+3% for vehicle and +17% for sonar), but in comparison with the corresponding approach, the size of the rule base is dramatically decreased (-71% for vehicle and -85% for sonar). Using the GA, here again, we have achieved a good accuracy-complexity trade-off.

In general, the number of selected classifiers is very small (less than 18 for vehicle and less than 8 for sonar), while keeping the same order of accuracy than the corresponding full 50 FRBCS ensembles. The GA is even able to decrease the test error in many cases (for instance, for all the configurations used with vehicle, but with 7 labels and 5 attributes, and for almost all the configurations used with sonar with 3 labels and 3 attributes).

In comparison with our previous study (i.e. without a high quality feature selection method), we are now able to improve the results obtained in the pima problem (previously, the best test errors obtained were 0.233 with a single classifier, 0.253 with an ensemble and 0.244 using the GA): now, not only an original ensemble, but also a selected ensemble, are outperforming the previous best results (0.232 for an ensemble generated with the greedy feature selection method and 0.233 using GRASP with  $\tau = 0.25$  and the GA).

In summary, combining bagging and the GA selection process 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. Now, combining these two techniques with

**Table 2. Results for the single FRBCSs with feature selection**

		Greedy				GRASP $\tau = 0.25$				GRASP $\tau = 0.50$				Random Subspace			
		Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar
3 labels 3 #attr.	5x2-cv	0.266	0.500	0.582	0.276	0.267	0.496	0.585	0.263	0.315	0.490	0.535	0.259	0.336	0.517	0.490	0.279
	#rules	1345	1235	1080	1240	1299	1177	1108	1245	1248	1178	1186	1230	1212	1083	1223	1248
	time	1.60	0.83	2.68	4.42	2.01	1.25	3.15	4.89	2.47	1.71	3.59	5.30	2.89	2.12	4.06	5.74
5 labels 3 #attr.	5x2-cv	<b>0.233</b>	0.434	0.422	0.283	<b>0.233</b>	0.415	0.465	0.247	0.281	0.481	0.427	0.243	0.296	0.512	0.414	0.247
	#rules	3910	3110	2310	3735	3762	2789	2746	3983	3613	2814	3199	3956	3492	2372	3717	4048
	time	2.98	1.49	4.18	4.82	3.37	1.86	4.88	5.26	3.71	2.31	5.50	5.67	4.12	2.68	6.14	6.10
7 labels 3 #attr.	5x2-cv	0.239	0.490	0.419	0.273	0.238	0.427	0.408	<b>0.229</b>	0.272	0.421	0.373	<b>0.200</b>	0.283	0.429	0.357	0.213
	#rules	7770	4905	4005	7005	7454	4348	4845	7655	7104	4483	5868	7662	6750	3754	7330	7882
	time	5.05	2.67	7.00	5.30	5.39	3.01	7.95	5.85	5.72	3.41	8.73	6.33	5.95	3.78	9.61	6.72
3 labels 5 #attr.	5x2-cv	0.266	0.446	0.549	<b>0.261</b>	0.263	0.448	0.512	0.240	0.271	0.448	0.478	0.229	0.280	0.445	0.444	0.231
	#rules	8925	6765	6820	7330	9053	6876	7337	8401	8770	6768	7893	8538	8510	6374	8501	8915
	time	7.24	3.39	10.30	5.72	7.92	3.89	11.15	6.41	8.15	4.25	11.74	6.82	8.58	4.61	12.49	7.29
5 labels 5 #attr.	5x2-cv	0.246	<b>0.376</b>	0.430	0.287	0.240	<b>0.406</b>	0.407	0.234	<b>0.244</b>	0.410	0.390	0.209	<b>0.251</b>	0.420	0.375	0.210
	#rules	34135	14550	21880	30760	34228	15698	26766	35037	33084	15860	28467	36408	32257	14640	34446	39090
	time	39.77	25.45	64.92	14.93	40.44	25.60	68.16	15.95	39.90	25.92	69.73	16.62	39.94	25.97	73.33	17.37
7 labels 5 #attr.	5x2-cv	0.262	0.414	<b>0.402</b>	0.291	0.251	0.412	<b>0.363</b>	0.231	0.260	<b>0.392</b>	<b>0.337</b>	0.213	0.260	<b>0.395</b>	<b>0.320</b>	<b>0.204</b>
	#rules	79985	21560	51045	60915	79319	23718	59619	69140	76253	24424	62233	71557	74369	23009	79232	76650
	time	157.21	118.77	307.60	47.78	157.47	118.34	302.83	47.51	155.72	118.56	305.84	49.23	154.55	117.85	317.42	50.91

**Table 3. Results for the FRBCS ensembles**

		Bagging+Greedy				Bagging+GRASP $\tau = 0.25$				Bagging+GRASP $\tau = 0.50$				Bagging + Random Subspace			
		Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar
3 labels 3 #attr.	5x2-cv	0.265	0.505	0.583	0.252	0.265	0.489	0.576	0.250	0.285	0.496	0.545	0.258	0.331	0.515	0.487	0.281
	#rules	1310	1187	1076	1221	1276	1156	1112	1219	1243	1130	1154	1205	1191	1043	1193	1184
	avg. #rules time	26.20	23.74	21.52	24.42	25.53	23.12	22.25	24.38	24.86	22.59	23.09	24.09	23.81	20.87	23.86	23.67
5 labels 3 #attr.	5x2-cv	<b>0.229</b>	0.420	0.427	0.250	<b>0.232</b>	0.428	0.443	0.237	0.252	0.457	0.424	0.238	0.281	0.480	0.424	0.255
	#rules	3537	2721	2490	3588	3458	2613	2731	3738	3358	2509	3085	3683	3238	2156	3472	3608
	avg. #rules time	70.74	54.42	49.79	71.76	69.17	52.26	54.62	74.76	67.15	50.19	61.70	73.66	64.76	43.13	69.45	72.15
7 labels 3 #attr.	5x2-cv	0.236	0.447	0.400	<b>0.236</b>	0.236	0.423	0.389	<b>0.217</b>	0.249	0.408	0.379	<b>0.202</b>	0.272	0.436	0.370	0.223
	#rules	6845	4217	4167	6642	6665	4044	4791	7045	6361	3796	5533	6895	6042	3285	6567	6708
	avg. #rules time	136.91	84.35	83.34	132.84	133.30	80.89	95.81	140.90	127.23	75.91	110.66	137.89	120.84	65.69	131.35	134.16
3 labels 5 #attr.	5x2-cv	0.265	0.469	0.520	0.244	0.267	0.438	0.495	0.233	0.275	0.456	0.477	0.212	0.282	0.465	0.458	0.249
	#rules	8539	6220	6909	7248	8544	6355	7331	7953	8318	6159	7744	7952	8092	5721	8078	8151
	avg. #rules time	170.78	124.40	138.17	144.96	170.89	127.10	146.62	159.06	166.37	123.18	154.89	159.04	161.83	114.43	161.57	163.02
5 labels 5 #attr.	5x2-cv	0.232	<b>0.392</b>	0.395	0.241	0.238	0.413	0.391	<b>0.217</b>	<b>0.247</b>	0.426	0.388	0.220	<b>0.253</b>	0.427	0.374	<b>0.200</b>
	#rules	29284	12879	22269	26826	29525	13828	25220	30345	28646	12968	26780	30721	27590	12228	30756	31833
	avg. #rules time	585.68	257.57	445.38	536.53	590.51	276.57	504.40	606.89	572.91	259.37	535.61	614.42	551.80	244.56	615.13	636.66
7 labels 5 #attr.	5x2-cv	0.246	0.419	<b>0.378</b>	0.262	0.247	<b>0.408</b>	<b>0.358</b>	0.239	0.253	<b>0.406</b>	<b>0.338</b>	0.229	0.266	<b>0.389</b>	<b>0.331</b>	0.275
	#rules	64707	19016	49516	50284	65087	20223	54596	55234	63845	19050	59027	55449	61112	17837	66911	57658
	avg. #rules time	1294	380.32	990.32	1006	1302	404.46	1092	1105	1277	381.01	1181	1109	1222	356.74	1338	1153
		153.20	119.91	300.13	46.81	148.88	116.56	293.13	46.58	154.13	120.95	312.22	50.27	145.77	117.28	309.93	45.06

**Table 4. Results for the FRBCS ensembles selected by the GA, using the Training Error fitness**

		Bagging+Greedy				Bagging+GRASP $\tau = 0.25$				Bagging+GRASP $\tau = 0.50$				Bagging + Random Subspace			
		Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar	Pima	Glass	Vehicle	Sonar
3 labels 3 #attr.	5x2-cv	0.255	0.399	0.485	<b>0.238</b>	0.254	<b>0.368</b>	0.460	0.238	0.251	<b>0.356</b>	0.449	0.239	0.261	0.386	0.435	0.243
	#classifiers	3.4	6.6	9.1	12.0	5.7	11.1	7.8	15.5	4.8	15.7	11.1	17.5	3.2	13.5	13.9	18.0
	avg. #rules time	89.2	154.1	204.3	295.2	145.2	251.7	179.8	379.3	121.0	355.5	263.9	426.3	79.5	283.2	337.9	430.2
5 labels 3 #attr.	5x2-cv	<b>0.232</b>	0.378	0.393	0.264	<b>0.233</b>	0.383	0.386	0.256	<b>0.243</b>	0.392	0.389	0.236	<b>0.247</b>	0.411	0.394	0.240
	#classifiers	6.6	10.0	9.3	14.1	8.0	18.0	9.9	21.2	9.8	12.6	13.9	18.9	9.0	13.8	16.7	22.8
	avg. #rules time	469.9	546.0	494.1	1047.8	555.7	943.1	558.8	1605.9	662.5	648.5	876.2	1418.0	597.2	643.7	1216.9	1687.9
7 labels 3 #attr.	5x2-cv	0.243	0.406	0.375	0.244	0.239	0.380	<b>0.368</b>	0.253	0.261	0.406	0.363	0.244	0.253	<b>0.369</b>	0.360	0.285
	#classifiers	9.9	9.8	10.1	14.3	8.1	13.9	17.1	9.3	11.2	15.9	16.1	10.8	10.6	13.7	18.6	11.4
	avg. #rules time	1358.5	824.2	939.1	1989.5	1069.3	1146.8	1732.2	1406.9	1463.5	1247.8	1925.3	1594.6	1308.7	968.0	2705.8	1625.3
3 labels 5 #attr.	5x2-cv	0.255	<b>0.354</b>	0.461	0.244	0.251	0.371	0.451	<b>0.222</b>	0.256	0.359	0.427	<b>0.222</b>	0.258	0.375	0.419	<b>0.221</b>
	#classifiers	3.3	8.8	12.6	10.3	5.4	9.4	11.8	14.9	5.2	13.2	9.9	15.6	4.6	10.0	13.4	19.8
	avg. #rules time	176.4	126.9	144.5	148.3	171.0	125.1	153.2	164.8	172.0	122.9	158.9	163.5	165.2	114.5	166.9	164.4
5 labels 5 #attr.	5x2-cv	0.239	0.373	0.390	0.240	0.245	0.373	0.372	0.237	0.249	0.381	0.370	0.247	0.255	0.379	0.373	0.262
	#classifiers	9.9	10.9	12.7	9.1	12.7	13.4	12.1	7.2	14.1	12.2	13.6	8.0	13.8	13.7	15.2	6.8
	avg. #rules time	5829.1	2853.7	6082.3	5125.5	7435.5	3793.1	6968.9	4658.1	8137.4	3354.7	8171.8	5238.0	7737.5	3388.3	10810.5	4540.0
7 labels 5 #attr.	5x2-cv	0.253	0.398	<b>0.373</b>	0.266	0.257	0.393	<b>0.368</b>	0.277	0.263	0.388	<b>0.340</b>	0.299	0.271	0.389	<b>0.335</b>	0.297
	#classifiers	13.2	11.5	15.5	6.0	18.3	10.1	13.0	5.1	18.6	12.6	14.8	4.6	14.8	14.1	20.3	5.5
	avg. #rules time	17562.1	4640.3	17500.8	6373.3	24337.5	4291.3	18473.0	5968.5	23936.1	5061.8	21532.6	5590.5	18787.0	5312.1	31294.8	6732.6
		111.95	41.71	1129.2	1062.1	1333.5	435.0	1421.9	1178.1	1288.5	398.9	1470.5	1230.2	1269.0	383.4	1556.7	1230.4
						110.17	31.80	124.62	30.21	110.55	31.69	126.40	29.81	110.67	31.93	126.42	29.40

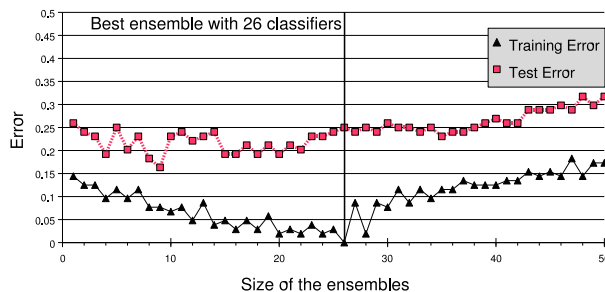
an advanced feature selection process improves also the accuracy on the smaller datasets (pima and glass).

## 6 On the different FRBCS ensembles contained in the best chromosome

Due

ical representation of the training and test error trends of the 50 ensembles selected by the best chromosome generated for the sonar dataset, with the bagging+random subspace approach and for a granularity of 3 and 5 attributes, is shown. The chosen solution (the one with the lowest training error  $TE=0$ , with 26 classifiers) is highlighted. Notice that the ensemble of 9 classifiers has a better test error and is actually smaller; and how bigger ensembles lead to bigger training and test errors.

**Figure 1. Training and test errors for the 50 ensembles selected by a chromosome**



## 7 Conclusion and future works

We have proposed the use of bagging and feature selection approaches like random subspace and greedy and GRASP-based Battiti's methods, together with a training error-guided multicriteria GA, to design FRBCS ensembles with a good accuracy-complexity trade-off, able to deal with classification problems with a large number of features. The results obtained in some popular data sets of high dimensions are quite promising.

Now, our future work will be concentrated on larger datasets (more than 1,000 examples), on the design of more advanced genetic ensemble selection techniques (for example, the use of Pareto-based algorithms), on the design of ensembles of more powerful FRBCSs, or even on the use of different/complementary fuzzy classifiers.

## References

[1] L. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley; 2004.

[2] O. Cordón, A. Quirin, L. Sanchez, A First Study on Bagging Fuzzy Rule-based Classification Systems with Multicriteria Genetic Selection of the Component Classifiers, in *2008 IEEE Intl. Workshop on GEFS*, Germany, 2008, pp. 11-16.

[3] L. Breiman, Bagging Predictors, *Machine Learning*, vol. 24:2, 1996, pp. 123-140.

[4] T. Ho, The Random Subspace Method for Constructing Decision Forests, *IEEE Trans. on PAMI*, vol. 20:8, 1998, pp. 832-844.

[5] H. Ishibuchi, T. Nakashima, M. Nii, *Classification and Modeling With Linguistic Information Granules*, Springer; 2005.

[6] R. Battiti, Using Mutual Information for Selecting Features in Supervised Neural Net Learning, *IEEE Trans. on Neural Networks*, vol. 5:4, 1994, pp. 537-550.

[7] T. A. Feo, M. G. C. Resende, Greedy Randomized Adaptive Search Procedures, *J. of Global Optimization*, vol. 6, 1995, pp. 109-133.

[8] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten Years of Genetic Fuzzy Systems: Current Framework and New Trends, *Fuzzy Sets and Systems*, vol. 141:1, 2004, pp. 5-31.

[9] R. Schapire, The Strength of Weak Learnability, *Machine Learning*, vol. 5:2, 1990, pp. 197-227.

[10] A. Tsymbal, M. Pechenizkiy, P. Cunningham, Diversity in Search Strategies for Ensemble Feature Selection, *Information Fusion*, vol. 6, 2005, pp. 83-98.

[11] L. Breiman, Random Forests, *Machine Learning*, vol. 45:1, 2001, pp. 5-32.

[12] D. Optiz and R. Maclin, Popular Ensemble Methods: An Empirical Study, *J. of Artificial Intelligence Research*, vol. 11, 1999, pp. 169-198.

[13] R.E. Banfield et al., A Comparison of Decision Tree Ensemble Creation Techniques, *IEEE Trans. on PAMI*, vol. 29:1, 2007, pp. 173-180.

[14] C. E. Shannon, W. Weaver, *The Mathematical Theory of Communication*, Univ. of Illinois Press, 1949.

[15] C.A. Coello, G.B. Lamont, D.A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd Edition*, Springer; 2007.

[16] T.G. Dietterich, Approximate Statistical Test for Comparing Supervised Classification Learning Algorithms, *Neural Computation*, vol. 10:7, 1998, pp. 1895-1923.