A First Study on Bagging Fuzzy Rule-based Classification Systems with Multicriteria Genetic Selection of the Component Classifiers

Oscar Cordón, Arnaud Quirin and Luciano Sánchez

Abstract— **Fuzzy rule-based classification systems (FRBCSs) are able to design interpretable classifiers but suffer from the curse of dimensionality when dealing with complex problems with a large number of features. In this contribution we explore the use of popular approaches for designing ensembles of classifiers in the machine learning field, bagging and random subspace, to design FRBCS multiclassifiers from a basic, heuristic fuzzy classification rule generation method, aiming to both improve their accuracy and to make them able to deal with high dimensional classification problems. Besides, a multicriteria genetic algorithm is proposed to select the component classifiers in the ensemble guided by the cumulative likelihood in order to look for an appropriate accuracy-complexity trade-off.**

I. INTRODUCTION

It is well known that fuzzy rule-based classification systems (FRBCSs) have the key advantage of their comprehensibility for human beings but suffer from the curse of dimensionality [1]. Apart from fuzzy rule selection/reduction methods, the most common solution is to consider a feature selection mechanism [2], either before the fuzzy system derivation or embedded in its design method.

In particular, there are different approaches based on the use of evolutionary algorithms (EAs) to put that into effect [3]. The flexibility provided by EAs to encode different structures to be learnt has made *genetic fuzzy systems* (GFSs) become one of the most successful approaches to hybridize fuzzy systems with learning and adaptation methods in the last fifteen years [4], being somehow able to deal with the resulting fuzzy system design interpretability-accuracy trade-off problem in a proper way [5]. Unfortunately, their role of fuzzy system identification methods make them inherit the same scalability problems and the design of GFSs capable of dealing with high dimensional problems with a large number of features and/or examples is still an open issue (and consequently a hot topic) [6].

On the other hand, in the last two decades there is an increasing interest on generating ensembles of classifiers in the machine learning community with the aim of improving the accuracy of a single classifier when solving a problem [7]. Boosting [8] and bagging [9] are the two most popular generic approaches to do so, as well as there are some other more recent proposals considering other ways to promote disagreement between the component classifiers, with feature selection being an extended strategy [10].

Up to now, boosting of fuzzy classifiers/rules has been already considered in some works (see Sec. II-B). However, the use of bagging is quite reduced and, up to our knowledge, no proposal has been made considering FRBCSs. In our opinion, that could be very interesting since, apart from the accuracy improvement, the use of bagging, and especially its combination with a feature selection process, can carry two interesting, collateral advantages for the design of FRBCSs, making them able to deal with the curse of dimensionality: i) the simplicity of the implicit parallelism of bagging, which allows for an easy parallel implementation; and ii) the problem partitioning due to the feature selection at the component classifier level, resulting in a tractable dimension for learning fuzzy rules for each individual classifier. These two ideas, along with the novelty of the use of the approach for FRBCSs, justify the development of the current study.

In this way, the current contribution aims to compose a first, preliminary approach to bagging FRBCSs with the final goal of allowing them to deal with high dimensional problems in a better way. To do so, we will start by applying three classifier ensemble design approaches (bagging, random subspace, and their combination) to the case of FRBCSs. A basic, quick, heuristic fuzzy classification rule generation method, belonging to Ishibuchi et al.'s family [1], will be considered to design the FRBCSs. Although we actually know the accuracy of the resulting multiclassifiers will be always limited by the use of this very simple and thus inaccurate learning method, we prefer considering it in this first study due to its quickness.

With the aim of both increasing the accuracy and reducing the complexity (thus increasing the interpretability) of the final classifier ensemble as much as possible, a multicriteria genetic algorithm (GA) for component classifier selection guided by the cumulative likelihood and based on the use of the lexicographic order will be proposed.

This paper is set up as follows. In the next section, the background required to understand our contribution is presented by reviewing popular classifier ensemble design approaches and previous works on fuzzy classifier ensembles. Sec. III introduces our approaches for designing FRBCS ensembles, while Sec. IV describes the proposed GA for component classifiers selection. The experiments developed and their analysis are shown in Sec. V. Finally, Sec. VI collects some concluding remarks and future research.

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

O. Cordón 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

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

II. BACKGROUND AND RELATED WORK

A. Classifier Ensemble Design Approaches

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. The best possible situation for an ensemble is that where the individual classifiers are both accurate and fully complementary, in the sense that they make their errors on different parts of the problem space [11], thus they rely for their effectiveness on the "instability" of the base learning algorithm.

According to the existing literature, there is a classical group of approaches to generate an classifier ensembles considering *data resampling* to obtain different training sets to derive each individual classifier. In *bagging* [9], 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 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 more diverse set of approaches which induct the individual classifier diversity using some ways different from resampling [12]. 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* [10], where each feature subset is randomly generated, is one of the most representative methods of this kind.

The interested reader is referred to [11], [13] for two outstanding reviews for the case of decision tree ensembles (both of them) and neural networks (the latter), including exhaustive experimental studies.

B. Previous Work on Fuzzy Classifier Ensembles

An early work of fuzzy classifier ensemble design, mainly focused on voting reasoning schemes, was proposed in [14]. The use of boosting for the design of fuzzy classifier ensembles has been considered in some works [15], [16], [17], [18]. However, only a few contributions for bagging fuzzy classifiers have been proposed considering fuzzy neural networks (together with feature selection) [19], fuzzy adaptive neural networks [15] and fuzzy clustering-based classifiers [20] as component classifier structures. Up to our knowledge, no proposal has been made considering FRBCSs.

Two advanced GFS-based contributions are worthy to be mentioned. On the one hand, an FRBCS ensemble design technique is proposed in [21] based on the use of some niching GA-based feature selection methods to generate the diverse component classifiers, and of another GA for classifier fusion by learning the combination weights. On the other hand, another interval and fuzzy rule-based ensemble design method based on the use of a single- and multiobjective genetic selection is introduced in [22], [23]. In this case, the coding scheme allows an initial set of either interval or fuzzy rules, considering the use of a different features in their antecedents, to be distributed among different component classifiers trying to make them as diverse as possible by means of two accuracy and one entropy measures. Besides, the same authors presented a previous proposal in [24], where a multiobjective EA generated a Pareto set of FRBCSs with different accuracy-complexity tradeoffs to be combined into an ensemble.

III. BAGGING FUZZY RULE-BASED CLASSIFICATION SYSTEMS

A. 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 $\ldots$ and x_n is A_{jn} then Class C_j with CF_j, $j = 1, 2, \ldots, N$, and they will take their decisions by means of the single-winner method [1].

To derive the fuzzy knowledge bases, one of the heuristic methods proposed by Ishibuchi et al. in [1] is considered: C_j is computed as the class h with maximum confidence according to the rule compatible training examples $D(A_j) = \{x_1, \ldots, x_m\}$:

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

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 [1]):

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

This method is good for our aim of designing FRBCS ensembles since it is simple and quick. However, it carries some drawbacks. The first one is its low accuracy, which will affect the final accuracy of the generated ensembles. Besides, it generates an excessive number of rules, which will make impossible to run it on pure bagging approaches without feature selection when the number of problem attributes and the granularity are high. To solve that, one can think on using the extension of the popular Wang and Mendel's method for classification problems [25], but the resampling applied by bagging would have a very small influence on the fuzzy rule base generated by the method (all the consequent classes would be the same and only the certainty degree would be affected).

B. FRBCS Ensemble Design Approaches Considered

Three different ensemble generation methods, bagging, random subspace, and their combination, are considered to build FRBCS multiclassifiers in the current study.

For the bagging approach, the bags are generated with the same size as the original training set, as commonly done. In the case of random subspace, all the classifiers will consider the same number of features, and they will be independently, randomly selected for each of them between those available. Finally, the bagging + random selection approach combines the previous two design mechanisms.

In order to keep the interpretability of the generated classifier ensembles as high as possible, two decisions will be 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 three approaches.

IV. A NEW PROPOSAL OF A MULTICRITERIA GENETIC-BASED CLASSIFIER ENSEMBLE SELECTION METHOD

The determination of the optimal size of an ensemble is an important issue for obtaining both the smallest test error and a good accuracy-complexity trade-off. In the current contribution we aim to propose a GA-based classifier selection method to be applied on the generated FRBCS ensemble. Some previous approaches have considered the use of training error measures, such as the ensemble training error [26], or the average of the individual classifier classification errors either on the training set or on their respective bags [23] for this task. However, these kinds of measures do not seem appropriate as they can lead to overfitting the data set, and they should be avoided or complemented with other diversity measures, as in the latter paper.

In this way, we propose to guide our GA by computing the cumulative likelihood of the selected ensemble, computed as described in the next subsections.

A. Likelihood-based Quality Assessment of a Subset of an Ensemble

We will use the likelihood to assess the quality of the selected subsets $S \subset T = \{1, \ldots, l\}$, as it allows us to discern differences between ensembles with the same training error and, especifically, between those with null error. In the latter cases, a training error-based measure guiding a learning process will automatically end up with the learning, while likelihood will go on improving the estimations of the probability distributions for each class, thus reducing the chances of overfitting the training data. Let the classes $h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_l(\mathbf{x})$ be the decisions of the component classifiers of the selected ensemble S for an input value $\mathbf{x} = (x_1, \ldots, x_n)$. 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 likelihood 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 ϵ foresees that case for which none of the members of the subset has found the true class of the pattern.

B. Cumulative Likelihood

The simplest strategy to find the optimal set S^* verifying $S^* = \arg\max_{S \subset T} L'_S$ is a greedy algorithm based on either starting from scratch and adding the best new members till the likelihood of the augmented ensemble is equal or lower (forward selection), or starting with the whole ensemble and iteratively removing members until the likelihood starts to decrease (backward selection). Both greedy algorithms are $\mathcal{O}(l^2)$, but they will not find the global optimum but in particular cases.

We propose to sort the members of the ensemble, 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 degree to which a permutation fulfills this goal is measured by means of the *cumulative likelihood* of the ensemble, defined as the vector containing the likelihood values of those subsets comprising the first classifier, the set formed by the first and the second, and so on.

C. Genetic Optimization of the Cumulative Likelihood

The main novelty of our proposal is that it deals with component classifier selection as a multicriteria problem 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 likelihood), *from a single chromosome*.

To do so, the GA looks for an optimal ordering of the component classifiers. The coding scheme is thus based on an order-based representation, a permutation $\Pi = \{j_1, j_2, \ldots, j_l\}$ 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 on.

The fitness function is thus multicriteria, being composed of an array of l values, $L^i = L'_{\{j_1, j_2, \ldots, j_i\}}$, corresponding to the cumulative likelihood of the l mentioned ensemble designs. Thanks to all of this, the best chromosome is that member in the population whose maximum cumulative likelihood is the highest. Then, the output of the GA is the ensemble comprising the classifiers from the first one to the one having the maximum cumulative likelihood value (although any other design not having the optimal likelihood but, for example, showing a lowest complexity can also be directly extracted).

Instead of using a Pareto-based approach [27], a lexicographical order is considered to deal with the multicriteria optimization, since in our opinion it better matches our scenario. In this way, when comparing two chromosomes, one is better than the other if it takes a better (higher) maximum value of the cumulative likelihood. 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.

In order to increase its convergence rate, the algorithm 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 first study, we have considered OX crossover and the usual exchange mutation.

V. EXPERIMENTS AND ANALYSIS OF RESULTS

A. Experimental Setup

To evaluate the performance of the FRBCS ensembles generated, we have selected four popular data sets from the UCI machine learning repository (see Table I). In all of them, every attribute is continous. As can be seen, the number of features range from a small number (8) to a large one (60). However, the number of examples is somehow low, ranging from 208 to 846, and we have left for future works the study of other data sets with a larger number of examples (higher than 1,000), both with small and large numbers of attributes.

TABLE I
DATA SETS CONSIDERED

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

In order to compare the accuracy of the considered classifiers, we used Dietterichs 5×2-fold cross-validation (5×2-cv) [28]. Two different granularities, 3 and 5, are tested for the single FRBCS derivation method, as well as it is run by considering three different feature sets: the whole set[1] and two subsets of size 3 and 5 selected by means of the largely used Battiti's MIFS filter feature selection method [29], which has been run by considering a discretization of the real-valued attribute domains in ten parts and setting the β parameter to 1.

The FRBCS ensembles generated are initially comprised by 50 classifiers, before applying the genetic selection method. In order to make the learning problem tractable and to obtain compact component classifiers, small values for the granularity and the number of features selected (in the cases where random subspace is considered) has been chosen (3 and 5 in both cases).

[1]For Sonar and Vehicle, we have considered a subset of 10 features also selected with Battiti's method since the number of possible rules is too large to allow the run with all the attributes.

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.

B. Single FRBCS vs. Bagging FRBCS ensembles

The statistics (5×2-cv error, number of rules and run time required for each run, expressed in seconds) for the single FRBCSs are collected in Table II. There are three blocks for each granularity value considered, each of them with three rows corresponding to three feature sets sizes: the whole initial set, 3 and 5.

TABLE II
RESULTS FOR THE SINGLE FRBCSS

		Pima	Glass	Vehicle	Sonar
3 labels	5×2-cv	0.266	0.416	0.482	**0.250**
all #attr.	#rules	2497.50	2571.80	13811.60	8564.00
	time	3.06	5.39	45.98	5.87
3 labels	5×2-cv	0.266	0.500	0.582	0.276
3 attr.	#rules	26.90	24.70	21.60	24.80
	time	N/A	N/A	0.10	0.13
3 labels	5×2-cv	0.266	0.446	0.549	0.261
5 attr.	#rules	178.50	135.30	136.40	146.60
	time	0.17	N/A	0.24	0.16
5 labels	5×2-cv	0.272	0.394	**0.390**	0.285
all #attr.	#rules	11199.90	5535.10	67118.33	44040.33
	time	96.46	459.06	1424.47	701.01
5 labels	5×2-cv	**0.233**	0.434	0.422	0.283
3 attr.	#rules	78.20	62.20	46.20	74.70
	time	N/A	N/A	0.13	0.13
5 labels	5×2-cv	0.246	**0.376**	0.430	0.287
5 attr.	#rules	682.70	291.00	437.60	615.20
	time	0.82	0.51	1.37	0.35

On the other hand, those results for the FRBCS ensembles generated from the three different approaches considered, bagging, random subspace and their combination, are shown in Table III. No value is shown for Vehicle and Sonar with 5 labels, due to the excessive computation time required.

As can be seen in Table II, the best results are achieved with 5 labels in every problem but in Sonar, where having a larger granularity overfits the data set. Besides, it seems a feature selection is always needed to achieve the best result, although this conclusion can not be fully checked for the case of the two larger problems, Vehicle and Sonar.

Concerning the FRBCS ensembles obtained, the first conclusion we can draw is that bagging is performing properly as it reduces the test error in every case (this confirms our assumption that the classifiers are unstable enough). Nevertheless, this error decrease is obtained at the cost of a huge complexity addition. We were expecting to get a good trade-off between accuracy and complexity through the two ensemble design techniques considering feature selection, random subspace and bagging + random subspace. In fact, when applying these approaches, either a lower number of rules to that of the individual FRBCS with all the variables is usually obtained or the learning problem

TABLE III

RESULTS FOR THE FRBCS ENSEMBLES

		Pima	Glass	Vehicle	Sonar
Bagging 3 labels	5×2-cv	0.265	0.405	0.407	**0.166**
	#rules	112393	104630	639175	619296
	time	148.29	256.18	2289.85	350.58
R.S. 3 labels	5×2-cv	0.333	0.494	0.499	0.284
	#rules	1222	1107	1204	1231
	time	1.40	0.70	1.69	0.75
R.S. 3 labels	5×2-cv	0.284	0.431	0.458	0.235
	#rules	8542	6358	8437	8902
	time	7.39	3.68	10.07	2.80
Bag. + R.S. 3 attr.	5×2-cv	0.331	0.534	0.500	0.284
	#rules	1201	1038	1189	1196
	time	1.33	0.74	1.74	0.72
Bag. + R.S. 5 attr.	5×2-cv	0.285	0.454	0.451	0.241
	#rules	7983	5709	8016	8122
	time	6.97	3.80	9.92	2.64
Bagging 5 labels	5×2-cv	0.269	**0.364**	N/A	N/A
	#rules	450291	213365	N/A	N/A
	time	556.92	254.49	N/A	N/A
R.S. 5 labels	5×2-cv	0.288	0.500	0.425	0.267
	#rules	3511	2493	3661	4024
	time	2.89	1.59	3.91	1.55
R.S. 5 labels	5×2-cv	**0.253**	0.418	0.380	0.201
	#rules	31910	15162	33951	38921
	time	35.81	23.47	68.77	12.46
Bag. + R.S. 3 attr.	5×2-cv	0.285	0.486	0.429	0.278
	#rules	3219	2182	3443	3623
	time	2.81	1.64	3.88	1.48
Bag. + R.S. 5 attr.	5×2-cv	0.256	0.423	**0.375**	0.212
	#rules	27719	12330	30647	32031
	time	34.29	23.08	67.22	11.66

becomes tractable when it was not for the latter. Besides, the run times are significantly reduced. Unfortunately, the two latter techniques show worse performance than pure bagging for every problem when using 3 labels, and also for Glass with 5 labels. Moreover, they do not always overcome their counterpart single classifier in the two smaller problems. It seems that, opposite to the use of random subspace with decision trees, the fuzzy classifier induction technique considered (see Sect. III-A) is not able to derive accurate classifiers when using randomly selected features and that a heuristic selection as that applied by Battiti's MIFS based on mutual information is required. This will be studied in future works.

In the two feature selection approaches, the ensembles considering 5 features always outperform the ones with 3, as expected. However, bagging + random subspace variants do not always overcome their pure random subspace counterparts.

Finally, notice that overall, the best FRBCS classifier ensembles generated for Glass, Vehicle and Sonar get a lower error than their corresponding best individual fuzzy classifier. For the case of Pima, the best ensemble is outperformed by the best single FRBCS, showing how our ensemble designs are not beneficial for this data set.

TABLE IV

RESULTS FOR THE FRBCS ENSEMBLES SELECTED BY THE GA

		Pima	Glass	Vehicle	Sonar
Bagging 3 labels all #attr.	5×2-cv	0.257	0.363	0.402	**0.198**
	#classifiers	5.3	8.5	11.8	6.8
	#rules	11848	17975	161038	88489
	avg. #rules	2241	2140	13586	13016
	time	105.44	30.36	125.11	29.66
Random Subspace 3 labels 3 attr.	5×2-cv	0.265	0.388	0.446	0.259
	#classifiers	2.6	10.3	12.0	12.6
	#rules	64	227	285	316
	avg. #rules	24	22	23	25
	time	105.44	29.97	118.81	29.09
Random Subspace 3 labels 5 attr.	5×2-cv	0.260	0.380	0.426	0.216
	#classifiers	2.4	9.9	11.3	17.5
	#rules	425	1236	1980	3142
	avg. #rules	176	124	175	183
	time	106.68	29.95	118.62	29.14
Bagging + Random S. 3 labels 3 attr.	5×2-cv	0.260	0.388	0.440	0.236
	#classifiers	3.4	13.7	11.1	17.5
	#rules	83	287	271	421
	avg. #rules	24	21	24	24
	time	106.55	29.75	119.37	28.89
Bagging + Random S. 3 labels 5 attr.	5×2-cv	0.253	0.385	0.421	0.235
	#classifiers	4.4	12.7	10.7	17.4
	#rules	719	1471	1789	2877
	avg. #rules	163	115	169	166
	time	106.06	30.25	119.46	28.98
Bagging 5 labels all #attr.	5×2-cv	0.271	**0.354**	N/A	N/A
	#classifiers	14.4	17.9	N/A	N/A
	#rules	129676	77205	N/A	N/A
	avg. #rules	9045	4313	N/A	N/A
	time	105.46	30.45	N/A	N/A
Random Subspace 5 labels 3 attr.	5×2-cv	**0.244**	0.431	0.391	0.246
	#classifiers	5.0	9.3	16.1	15.2
	#rules	367	523	1228	1267
	avg. #rules	73	56	76	84
	time	105.41	30.24	118.93	29.05
Random Subspace 5 labels 5 attr.	5×2-cv	0.253	0.383	0.378	0.280
	#classifiers	7.5	14.0	9.5	3.2
	#rules	4920	4298	8290	2867
	avg. #rules	653	312	877	903
	time	106.96	29.80	122.12	29.31
Bagging + Random S. 5 labels 3 attr.	5×2-cv	0.247	0.432	0.397	0.247
	#classifiers	9.0	10.5	12.4	22.8
	#rules	604	504	899	1654
	avg. #rules	67	50	73	72
	time	106.31	29.98	119.81	29.02
Bagging + Random S. 5 labels 5 attr.	5×2-cv	0.253	0.379	**0.359**	0.247
	#classifiers	9.7	13.1	12.8	13.4
	#rules	5532	3425	8995	8680
	avg. #rules	564	262	708	662
	time	105.65	30.04	122.54	29.32

C. Genetic Selection for FRBCS ensembles

The values for the different indices related to the finally selected FRBCS ensembles using the proposed GA are collected in Table IV. Apart from the usual test error and run time, the number of component classifiers, the total number of fuzzy rules in the selected ensemble and the mean of the number of rules in each classifier are shown for each case.

For comparison purposes, we have also run the same GA considering the usual ensemble training error as fitness function. The results were always similar or worse (especially for Sonar), and are not shown due to the lack of space.

In view of these results, we should first notice that the selected ensembles outperform the initial ones in all the cases but in three configurations for the Sonar problem (those with the best accuracy: pure bagging with 3 labels, and random subspace for 5 labels, with and without bagging). In those cases, the training errors of the individual classifiers are equal to zero, and it seems the number of generations of the GA was too small in order the cumulate likelihood measure achieves a good solution. On the other hand, it can also be seen how the accuracy-complexity trade-off is pretty good since the number of selected classifiers is usually small (around 10), being always lower than 23, less than the half of the original number of component classifiers.

Finally, comparing the best results overall, the selected ensembles are the best choice for two of the four problems, Glass and Vehicle. While in the former case this requires a huge complexity increase with respect to the best individual classifier in Table II, in the latter, the accuracy improvement also comes with a huge complexity decrease. As said, for the case of Pima, the FRBCS ensembles are not able to improve the single fuzzy classifiers. For Sonar, very good results in terms of both accuracy and complexity are obtained, significantly overcoming those of the single classifiers. Although the test errors of the best original FRBCS ensembles increase after the selection, as already mentioned, another good result is obtained. In summary, it seems that the proposed methodology performs better for the case of true high dimensional problems with a large number of variables, which was our original aim.

VI. CONCLUDING REMARKS AND FUTURE WORKS

We have proposed the use of bagging and random subspace approaches, together with a likelihood-guided multicriteria GA for component classifier selection, 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 dimension have been quite promising.

Many different future works arise from this first, preliminary study. We can mention the use of more advanced feature selection approaches for the component classifiers, the design of more advanced genetic ensemble selection techniques (for example, the use of Pareto-based algorithms), or the use of more powerful FRBCS learning methods.

REFERENCES

[1] H. Ishibuchi, T. Nakashima, M. Nii, *Classification and Modeling With Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*, Springer; 2005.

[2] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer; 1998.

[3] J. Casillas, O. Cordón, M.J. del Jesus, F. Herrera, Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems, *Information Sciences*, vol. 136:1-4, 2001, pp. 135-157.

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

[5] J. Casillas, F. Herrera, R. Pérez, M.J. del Jesus, P. Villar (Eds.), Special Issue on Genetic Fuzzy Systems and the Interpretability-accuracy Trade-off, *International Journal of Approximate Reasoning*, vol. 44:1, January 2007.

[6] 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.

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

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

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

[10] T. Ho, The Random Subspace Method for Constructing Decision Forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20:8, 1998, pp. 832-844.

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

[12] Z.H. Zhou, Ensembling local learners through multimodal perturbation, *IEEE Transactions of Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35:4, 2005, pp. 725-735.

[13] R.E. Banfield, L.O. Hall, K.W. Bowyer, W.P. Kegelmeyer, A Comparison of Decision Tree Ensemble Creation Techniques, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29:1, 2007, pp. 173-180.

[14] H. Ishibuchi, T. Nakashima, and T. Morisawa, Voting in fuzzy rule-based systems for pattern classification problems, *Fuzzy Sets and Systems*, vol. 103:2, 1999, pp. 223-238.

[15] W. Pedrycz, A. Breuer, N.J. Pizzi, Fuzzy Adaptive Logic Networks as Hybrid Models of Quantitative Software Engineering, *Intelligent Automation and Soft Computing*, vol. 12:2, 2006, pp 189-209.

[16] W. Pedrycz and K.C. Kwak, Boosting of granular models, *Fuzzy Sets and Systems*, vol. 157:22, 2006, pp. 2934-2953.

[17] M.J. del Jesus, F. Hoffmann, L.J. Navascues, L. Sánchez, Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms, *IEEE Transactions on Fuzzy Systems*, vol. 12:3, 2004, pp. 296-308.

[18] J. Otero, L. Sanchez, Induction of descriptive fuzzy classifiers with the Logitboost algorithm, *Soft Computing*, vol. 10:9, 2006, pp. 825-835.

[19] H. Takahashi and H. Honda, Lymphoma prognostication from expression profiling using a combination method of boosting and projective adaptive resonance theory, *Journal of Chemical Engineering of Japan*, vol. 39:7, 2006, pp. 767-771.

[20] C.-A. Tsai, T.-C. Lee, I.-C. Ho, U.-C. Yang, C.-H. Chen, J.J. Chen, Multi-class clustering and prediction in the analysis of microarray data, *Mathematical Biosciences*, vol. 193:1, 2005, pp. 79-100.

[21] J.J. Aguilera, M. Chica, M.J. del Jesus, F. Herrera, Niching Genetic Feature Selection Algorithms Applied to the Design of Fuzzy Rule Based Classification Systems, *in 2007 IEEE Intl. Conference on Fuzzy Systems (FUZZ-IEEE 2007)*, London, UK, 2007, pp. 1794-1799.

[22] Y. Nojima and H. Ishibuchi, Designing fuzzy ensemble classifiers by evolutionary multiobjective optimization with an entropy-based diversity criterion, *in Sixth International Conference on Hybrid Intelligent Systems and Fourth Conference on Neuro-Computing and Evolving Intelligence*, 2006, CD-ROM, 4 pages.

[23] Y. Nojima and H. Ishibuchi, Genetic rule selection with a multi-classifier coding scheme for ensemble classifier design, International Journal of Hybrid Intelligent Systems, 2007, in press.

[24] H. Ishibuchi and Y. Nojima, Evolutionary multiobjective optimization for the design of fuzzy rule-based ensemble classifiers, *International Journal of Hybrid Intelligent Systems*, vol. 3:3, 2006, pp. 129-145.

[25] Z. Chi, H. Yan, T. Pham, *Fuzzy algorithms with applications to image processing and pattern recognition*, World Scientific; 1996.

[26] D. Ruta and B. Gabrys, Classifier selection for majority voting, *Information Fusion*, vol. 6:1, 2005, pp. 63-81.

[27] C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer; 2002.

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

[29] R. Battiti, Using mutual information for selecting features in supervised neural net learning, *IEEE Transactions on Neural Networks*, vol. 5:4, 1994, pp. 537-550.