A First Study on a Fuzzy Rule-Based Multiclassification System Framework Combining FURIA with Bagging and Feature Selection

Krzysztof Trawiński, Oscar Cordón, and Arnaud Quirin

Abstract— **In this work, we conduct a preliminary study considering a fuzzy rule-based multiclassification system design framework based on Fuzzy Unordered Rule Induction Algorithm (FURIA). This advanced method serves as the fuzzy classification rule learning algorithm to derive the component classifiers considering bagging combined with feature selection. We develop a study on the use of both bagging and feature selection to design a final FURIA-based fuzzy multiclassifier applied to ten popular UCI datasets. The results obtained show that this approach provides promising results.**

I. Introduction

Multiclassification systems (MCSs) (also called multiclassifiers or classifier ensembles) 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 recently using different kinds of fuzzy classifiers [2], [3], [4], [5], [6], [7], [8].

Fuzzy Unordered Rule Induction Algorithm (FURIA) [9], [10] is a powerful fuzzy classification rule learning algorithm that can deal with a very common problem of fuzzy rule-based classification systems (FRBCSs), the so-called curse of dimensionality [11]. By combining advantages of the RIPPER algorithm [12] with fuzzy logic, this algorithm is able to generate simple and compact sets of fuzzy classification rules, even when tackling datasets with a large amount of features. Apart from its ability to deal with high dimensional datasets, this approach has shown a performance advantage in comparison to classical machine learning methods such like RIPPER [12] and C4.5 [13].

There are several techniques in order to obtain diversity, which leads to a highly accurate ensemble [1], [14], among the classifiers. Bagging [15] and boosting [16] are the two most popular generic approaches to do so [17]. There are also other more recent proposals considering other ways to promote disagreement between the component classifiers, with feature selection being an extended strategy [18]. All in all, it turned out that a combination between bagging and feature selection is a generic approach leading to good MCS designs for any kind of classifier learning method [19].

The idea of this paper is a preliminary study of the performance of FURIA-based fuzzy MCSs. FURIA-based fuzzy MCSs are build using a combination of bagging and feature selection. We considered three different types of feature selection algorithms: random subspace [18], mutual information-based feature selection (MIFS) [20], and the random-greedy feature selection based on MIFS and the GRASP approach [21].

In order to test the accuracy of the proposed fuzzy MCSs, we conduct experiments with 10 datasets taken from the UCI machine learning repository and provide a study of the results obtained. Then, we compare them against single FURIA classifiers.

This paper is structured as follows. The next section presents a state of the art about MCSs and fuzzy MCSs. In Sec. III the FURIA algorithm is described, while Sec. IV describes our approach for designing FURIA-based fuzzy MCSs. The experiments developed and their analysis are detailed in Sec. V. Finally, Sec. VI collects some concluding remarks and future research lines.

II. Background and related work

This section explores the current literature related to the generation of fuzzy rule-based multiclassification systems (FRBMCSs). The techniques used to generate MCSs and fuzzy MCSs are described in Sec. II-A and II-B, respectively.

A. Related work on MCSs

A MCS 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]. These kinds of methods have gained a large acceptance in the machine learning community during the last two decades due to their high performance. Decision trees are the most common classifier structure considered and much work has been done in the topic [22], [23], although they can be used with any other type of classifiers (the use of neural networks is also very extended, see for example [24]).

There are different ways to design a classifier ensemble. 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* [15], they are independently learnt from resampled training sets ("bags"), which are randomly selected with replacement from the original training data set. *Boosting* methods [16] sequentially generate the individual classifiers (weak learners) 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

Krzysztof Trawiński, Oscar Cordón, and Arnaud Quirin are with European Centre for Soft Computing. Mieres, Spain. (email: {krzysztof.trawinski, oscar.cordon, arnaud.quirin}@softcomputing.es).

selection probability 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, not based on resampling, which induct the individual classifier diversity [25]. Feature selection plays a key role in many of them where each classifier is derived by considering a different subset of the original features [14], [26]. *Random subspace* [18], where each feature subset is randomly generated, is one of the most representative methods of this kind.

Finally, there are some advanced proposals that can be considered as *combinations of the two groups*. The most extended one could be *random forests* [27], 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 [22], [24] for two surveys for the case of decision tree (both) and neural network ensembles (the latter), including exhaustive experimental studies.

B. Previous Work on Fuzzy MCSs

The use of boosting for the design of fuzzy classifier ensembles has been considered in some works, taking the weak learners as fuzzy variants of neural networks [8], [28]: as granular models [6], as neuro-fuzzy systems [29], as well as single fuzzy rules [30], [31], [32].

However, only a few contributions for bagging fuzzy classifiers have been proposed considering, fuzzy adaptive neural networks [28], fuzzy neural networks (together with feature selection) [33], fuzzy clustering-based classifiers [34], neuro-fuzzy systems [3], and fuzzy decision trees [2], [4] as component classifier structures.

Especially worth mentioning is the contribution of Bonissone et al. [2]. This approach hybridizes Breimann's idea of random forests [27] with fuzzy decision trees [35]. Such resulting fuzzy random forest combines characteristics of MCSs with randomness and fuzzy logic in order to obtain a high quality system joining robustness, diversity, and flexibility to not only deal with traditional classification problems but also with imperfect and noisy datasets. The results show that this approach obtains good performance in terms of accuracy for all the latter problem kinds.

In our previous studies [36], [37], [38], [39], we proposed a MCS methodology based on classical MCS design techniques such as bagging and feature selection with a fuzzy rule-based classification system (FRBCS) as a base classifier. The fuzzy classification rule learning algorithm considered was the basic heuristic method proposed by Ishibuchi [11]. A multicriteria genetic algorithm (GA) was used for a static component classifier selection from FRBMCSs guided by several fitness functions based on training error and likelihood, as well as bicriteria fitness functions based on training error and likelihood or diversity measures.

Some other contributions based on the use of GAs should also be remarked. On the one hand, an FRBCS ensemble design technique is proposed in [40] considering some niching GA-based feature selection methods to generate the diverse component classifiers, and another GA for classifier fusion by learning the combination weights. On the other hand, another interval and fuzzy rule-based ensemble design method using a single- and multiobjective genetic selection process is introduced in [5], [41]. In this case, the coding scheme allows an initial set of either interval or fuzzy rules, considering the use of 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 [42], where an EMO algorithm generated a Pareto set of FRBCSs with different accuracy-complexity tradeoffs to be combined into an ensemble.

III. FURIA

Fuzzy Unordered Rules Induction Algorithm (FURIA) [9], [10] is an extension of the state-of-the-art rule learning algorithm called RIPPER [12], having its advantages such like simple and comprehensible fuzzy rule base, and introducing new features. FURIA provides three different extensions of RIPPER: i) it applies unordered rule sets instead of rule lists, ii) it takes an advantage of fuzzy rules instead of crisp ones, and iii) it proposes a novel rule stretching method in order to manage uncovered examples. Below the said features of FURIA are reviewed.

A. Unordered rule base instead of the list of rules

The first extension of FURIA is the following. It deals with a standard unordered rule base (RB) instead of a decision list, as the latter provides one crucial disadvantage. Particularly, a list of rules favors a default class, that introduces a bias. Here, for each class, a set of rules is generated using the one-vs.-rest strategy. Thus, FURIA separates each class from the other classes. In consequence, there is no default rule and the order of the rules is not important.

However, this new approach has two drawbacks. The first one concerns a conflict which arises when having the same coverage of several rules from different classes. The second one may take place when an example is not covered by any of the rules. The first drawback is rather unlikely to occur, eventhough in case it occurs, it may be resolved easily. The latter issue is solved by introducing a novel rule stretching method as described below.

B. Fuzzification of the RIPPER rules

The fuzzification of the RIPPER (crisp) rules corresponds to the transformation of the crisp values into the fuzzy ones, that is fuzzy sets with trapezoidal membership functions. Based on the training set the best fuzzy interval is generated. Considering the intervals of the crisp rules I_i as the cores $[b_i, c_i]$ of the fuzzy rule, a learning process aims at determining the optimal size of the supports of each of the antecedents $[a_i, d_i]$. It must be pointed that only the subset D_T^i of the training set D_T that have not been already covered by any of the antecedents ($A_j \in FI_j$, $j \neq i$) is considered in order to build a single antecedent ($A_i \in I_i$):

$$\begin{aligned} D_T^i &= \{x = (x_1 ... x_k) \in D_T | FI_j(x_j) > 0 \text{ for all } j \neq i\} \\ &\subseteq D_T \end{aligned} \tag{1}$$

Then, the D_T^i is divided into two subsets, the positive subset D_{T+}^i and the negative subset D_{T-}^i. The following measure, called rule purity, is used in order to check the quality of the fuzzification:

$$pur = \frac{p_i}{p_i + n_i} \tag{2}$$

where

$$p_i = \sum_{x \in D_{T+}^i} \mu_{A_i}(x) \; ; \; n_i = \sum_{x \in D_{T-}^i} \mu_{A_i}(x)$$

The rule fuzzification procedure is greedy and it iterates over all antecedents calculating the best fuzzification in terms of purity (see equation 2). The candidate values for a are those values laying on the left side from b belonging to D_T^i, and are expressed as: $x_i | x = (x_1, ..., x_k) \in D_T^i, x_i < b$. The candidate values for d are those values laying on the right side from c belonging to D_T^i, and are expressed as: $x_i | x = (x_1, ..., x_k) \in D_T^i, x_i > c$. In case of a tie, the larger fuzzy set, the one having a larger distance from the core, is selected. Then, the antecedent with the highest purity value is selected to be fuzzified. The whole process ends up when all antecedents are fuzzified. This procedure is repeated only once, as it has been noticed that in almost all cases convergence is obtained after the first iteration.

C. Fuzzy classification rule structure and fuzzy reasoning method

Fuzzy rules of FURIA are composed of a class C_j and a certainty degree CD_j in the consequent, the most extended fuzzy classification rule structure [11], [43]. The final form of a rule is the following:

R_j : If x_1 is A_{j1} and ...and x_n is A_{jn}
then Class C_j with CD_j; $j = 1, 2, ..., N$.

The certainty degree of a given example x is defined as follows:

$$CD_j = \frac{2 \frac{|D_T^{C_j}|}{|D_T|} + \sum_{x \in D_T^{C_j}} \mu_r^{C_j}(x)}{2 + \sum_{x \in D_T} \mu_r^{C_j}(x)} \tag{3}$$

where $D_T^{C_j}$ stands for a subset of the training set in which the instances are affected to the class C_j. The fuzzy reasoning method used is the so-called voting-based method [43], [44]. In this approach, each fuzzy rule makes a vote for its consequent class. The vote strength of the rule is calculated as the product of the firing degree $\mu_r^{C_j}(x)$ and the certainty degree CD_j. The final decision given as the output is the class with the largest value of the accumulated vote, which is calculated as follows:

$$V_h = \sum_{\substack{R_j \in \text{RB} \\ C_j = h}} \mu_r^{C_j}(x) * CD_j \tag{4}$$

where h is the class for which the accumulated vote is computed. In this approach, all compatible fuzzy rules are responsible for the classification, which should provide a higher robustness. It must be pointed that when there is no rule of any class covering a given example x, a rule stretching procedure, explained in Sec. III-D, is executed.

D. Rule stretching

In case some examples of the training dataset not covered by any rule exist, a procedure, called rule stretching or rule generalisation, is applied. This algorithm enlarges the covering surface of the rules by deleting at least one antecedent from each of the rules. The generalization procedure aims to reach a minimal state i.e. only the minimal amount of antecedents are removed. In FURIA, rule stretching treats antecedents in the same order in which they were learned. Thus, it introduces implicitly a degree of importance among the antecedents, which decreases the complexity of the approach. The final list is then obtained by cutting the entire antecedents list at the point where an antecedent not satisfying a given example is encountered. To check that general rules are obtained, the following measure is used:

$$\frac{p+1}{p+n+2} \times \frac{k+1}{m+2}$$

where p and n are respectively the number of positive and negative examples covered by the rule, while m is the size of the entire antecedents list and k is the size of the generalized list. Note that the second part of the measure aims at discarding heavily pruned rules, as pruning is rather decreasing the relevance of the rule.

The interested reader is referred to [9], [10] for more details regarding the description of FURIA and its improvements with respect to the RIPPER algorithm.

IV. BAGGING FURIA-BASED FUZZY MCSS

In this section we will detail how the FURIA fuzzy MCSs 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 procedures in order to provide individual training sets (the so-called *bags*) to train FURIA classifiers. After the training, we get a FURIA-based fuzzy MCS, which is validated using the training and the test errors, as well as a measure of complexity based on the total number of component classifiers obtained from FURIA. The whole procedure is graphically presented in Fig. 1.

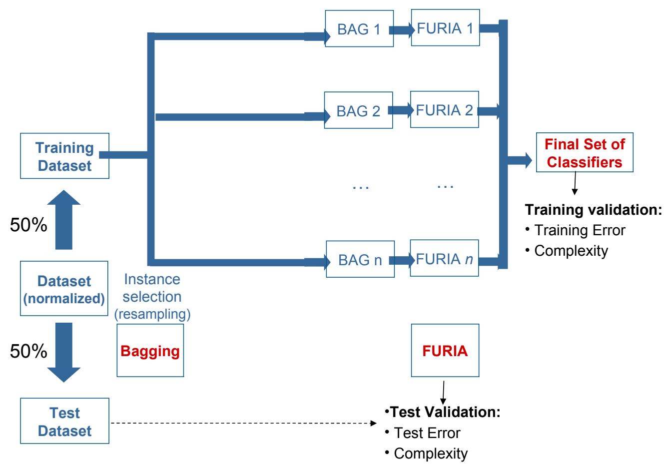

Fig. 1. Our framework: after the instance and the feature selection procedures, the component fuzzy classifiers are derived by the FURIA learning method. Finally, the output is obtained using a voting-based combination method.

A. FURIA-based fuzzy MCS design approaches

In [19] it was shown that a combination between bagging and feature selection composed a general design procedure which usually leads to good MCS designs, regardless the classifier structure considered. Hence, we decided to follow that idea and we integrate FURIA into a framework of that kind. We aim to combine the diversity induced by the MCS design methods and the robustness of the FURIA method in order to derive good performance fuzzy rule-based MCSs.

The term *bagging* is an acronym of bootstrap aggregation and refers to the first successful method proposed to generate MCSs [15]. This approach was originally designed for decision tree-based classifiers, however it can applied to any type of model for classification and regression problems. Bagging is based on bootstrap and consists of reducing the variance of the classification by averaging many classifiers that have individually been tuned to random samples that follow the sample distribution of the training set. The final output of the model is the most frequent value, called voting, of the learners considered. Bagging is the most effective when dealing with unstable classifiers, what means a small change in the training set can cause a significant change in the final model. In addition, it is recommended when a given dataset is composed of small amount of examples. Furthermore, bagging enables a parallel and independent learning of the learners in the ensemble.

In this contribution, the bags are generated with the same size as the original training set, as commonly done. Three different feature selection methods, random subspace [18], mutual information-based feature selection (MIFS) [20], and a random-greedy feature selection method based on MIFS and the GRASP approach [21], are considered.

Random subspace is a method in which a subset of features is randomly selected from the original dataset. Alternatively, the greedy Battiti's MIFS method is based on a forward greedy search using the mutual information measure [45], with regard to the class. This method orders a given set S of features by the information they bring to classify the output class considering the already selected features. The mutual information $I(C,F)$ for a given feature F is defined as:

$$I(C,F) = \sum_{c,f} P(c,f) \log \frac{P(c,f)}{P(c)P(f)} \qquad (5)$$

where $P(c)$, $P(f)$ and $P(c,f)$ are respectively the values of the density function for the class, the feature variables, and the joint probability density. In the MIFS method, a first feature f is selected as the one that maximizes $I(C,f)$, and then the features f that maximize $Q(f) = I(C,f) - \beta\sum_{s\in S} I(f,s)$ are sequentially chosen until S reaches the desired size. β is a coefficient to reduce the influence of the information brought by the already selected features.

The random-greedy 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 τ percent features according to the Q measure at each selection step. Parameter τ 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.

Random search such as random subspace for feature selection is a well-known approach in the multiclassifiers research field [2], [18], [23], [27], [46]. Nevertheless, the use of a heuristic such as a randomized variant of greedy Battiti's MIFS [20] combined with FURIA, which is a tree-based fuzzy rule generation approach, may lead to a performance improvement.

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

V. Experiments and Analysis of Results

This section presents all the experiments performed. Sec. V-A introduces the experimental setup. In Sec. V-B we check the good quality of single FURIA. Then, Sec. V-C shows results of FURIA-based fuzzy MCSs combined with bagging and feature selection, in which we compare all the feature selection approaches considered and report an advantage of our FURIA-based fuzzy MCS with bagging and feature selection over the single FURIA classifier.

A. Experimental setup

To evaluate the performance of the generated FURIA-based fuzzy MCSs, we have selected ten datasets with different characteristics concerning the number of examples, features, and classes from the UCI machine learning repository (see Table I). In order to compare the accuracy of the considered classifiers, we used Dietterich's 5×2-fold cross-validation (5×2-cv), which is considered to be superior to paired k-fold cross validation in classification problems [47].

TABLE I
DATASETS CONSIDERED.

abbrev.	dataset	#examples	#attr.	#classes
aba	abalone	4178	7	28
bre	breast	700	9	2
gla	glass	214	9	7
hea	heart	270	13	2
ion	ionosphere	352	34	2
let	letter	20000	16	26
mag	magic	19020	10	2
opt	optdigits	5620	64	10
pbl	pblocks	5474	10	5
pen	pendigits	10992	16	10

A feature subset, which are relative with respect to the initial size of features of the classification problem, is tested for the FURIA-based fuzzy MCSs using feature selection. The considered rule to select a feature subset size is following: if the size of an initial feature set is smaller than 10, then the feature subset size is equal to 5. If the size of an initial feature set is between 10 and 20, then the feature subset size is equal to 9. Finally, if a size of an initial feature set is larger than 30, then the Large feature subset size is roughly equal to 30% of the initial set (see Table II).

As described in Sec. IV-A, these features are to be selected by means of three different approaches: the greedy Battiti's MIFS filter feature selection method [20], the Battiti's method with GRASP (with τ equal to 0.5, see Sec. IV-A), and random subspace [18]. Battiti's method has been run by considering a discretization of the real-valued attribute domains in ten parts and setting the β coefficient to 0.1.

TABLE II
FEATURE SUBSET SIZES FOR EACH OF THE DATASETS CONSIDERED.

Dataset	#attr.	feat. subset size
abalone	7	5
breast	9	5
glass	9	5
heart	13	9
ionosphere	34	9
letter	16	9
magic	10	9
optdigits	64	18
pblocks	10	9
pendigits	16	9

The FURIA-based fuzzy MCSs generated are initially comprised by 3, 5, 7, and 10 classifiers in order to evaluate the impact of the ensemble size in the accuracy of the obtained MCS. A small number of component fuzzy classifiers (up to 10) is considered in this first study. Larger numbers are left for future works as well as the consideration of a classifier selection mechanism.

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.

B. Single FURIA-based fuzzy classifier for high dimensional problems

In the first place, we have conducted experiments on a single FURIA-based fuzzy classifier without feature selection in order to observe its behaviour on the different datasets selected. Notice that, some of them can be considered to be high dimensional, either with respect to the number of features or with respect to the number of examples.

We may observe that FURIA in isolation is able to deal with high dimensional datasets with many features (for instance optdigits, which has 64 features) as well as with many examples (for instance letter, which has 20.000 examples), providing good quality results (see Table III). Our aim in the reminder of this section is to check if the use of fuzzy MCSs based on FURIA allows us to improve the latter capability by obtaining a more accurate classification system.

TABLE III
RESULTS FOR A SINGLE FURIA-BASED FUZZY CLASSIFIER WITHOUT FEATURE SELECTION.

FURIA single classifier - All features										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.781	0.023	0.336	0.141	0.041	0.038	0.143	0.633	0.018	0.003
test err.	0.805	0.049	0.377	0.227	0.163	0.123	0.157	0.683	0.033	0.027

C. Combination of FURIA with bagging and feature selection

In this subsection, we present the results of the FURIA-based fuzzy MCSs obtained from the combination of bagging and the three feature selection algorithms considered (see Sec. IV-A).

This experiment is made with the aim to check if the combination of bagging with feature selection allows us to generate the most accurate ensembles as happened with other kinds of classifiers [36], [48].

Each table (Tables from IV to VI) presents a set of FURIA-based fuzzy MCSs with different ensemble sizes. The combination of each feature selection algorithm is shown in a different table.

We will do two types of analyses taking into account the test errors obtained. In the first analysis, we will compare the performance of the three different feature selection algorithms, and in the second we will benchmark the FURIA-based fuzzy MCS derived by the best previous feature selection approach against the single FURIA-based fuzzy classifier.

1) Feature selection approaches: in our first analysis, we are comparing the three different feature selection algorithms among them.

The results are presented in Tables IV, V, and VI.

- FURIA-based fuzzy MCSs considering bagging and the Greedy feature selection outperform the other approaches in 6 out of 10 cases (+1 tie), whereas FURIA-based fuzzy MCSs considering bagging and the Random-greedy feature selection do so 1 time (+2 ties) placing FURIA-based fuzzy MCSs considering bagging and the Random subspace feature selection approach in the last position with only 1 best accuracy value obtained (+1 tie), for ensembles composed of 3 classifiers.

- For ensembles composed of 5 classifiers, all the three approaches obtained the same results having 3 best accuracy values obtained (+1 tie).
- FURIA-based fuzzy MCSs considering bagging and the Random-greedy feature selection outperform the other approaches in 5 out of 10 cases, whereas FURIA-based fuzzy MCSs considering bagging and the Random subspace feature selection do so 3 times placing FURIA-based fuzzy MCSs considering bagging and the Greedy feature selection approach in the last position with only 2 best accuracy values obtained for ensembles composed of 7 classifiers.
- FURIA-based fuzzy MCSs considering bagging and the Random-greedy and the Random subspace feature selection obtained the best accuracy values in 4 out of 10 cases (+1 ties), whereas FURIA-based fuzzy MCSs considering bagging and the Greedy feature selection do so 1 times (+1 ties) for ensembles composed of 10 classifiers.
- Considering the overall results, FURIA-based fuzzy MCSs considering bagging and the Random-greedy and the Random subspace feature selection obtained the best overall accuracy values in 4 out of 10 cases (+1 ties), placing the Greedy feature selection in the last position with only 1 best accuracy value obtained (+1 tie).

TABLE IV

FURIA-BASED FUZZY MCSS WITH BAGGING AND GREEDY FEATURE SELECTION.

FURIA - Greedy feature selection										
3 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.657	0.020	0.163	0.095	0.039	0.053	0.115	0.511	0.015	0.015
test err.	0.769	0.051	0.360	0.199	0.161	0.124	0.140	0.664	0.031	0.049
5 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.642	0.018	0.123	0.090	0.040	0.039	0.114	0.499	0.014	0.011
test err.	0.762	0.047	0.348	0.196	0.157	0.111	0.139	0.654	0.031	0.044
7 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.625	0.017	0.116	0.073	0.038	0.034	0.114	0.502	0.014	0.009
test err.	0.756	0.044	0.337	0.187	0.153	0.104	0.139	0.643	0.030	0.043
10 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.622	0.017	0.114	0.074	0.035	0.029	0.116	0.501	0.014	0.008
test err.	0.753	0.045	0.335	0.184	0.147	0.100	0.140	0.639	0.030	0.041

In view of the results obtained it is rather hard to point out the best feature selection approach. Table VII summarizes the obtained results in the form of a summarized matrix showing the number of wins (W), ties (T), and loses (L) for three feature selection algorithms for each ensemble size. In view of the overall results, collected in the last row of the table, we can may be highlight the performance of the Random-Greedy feature selection approach to generate FURIA-based fuzzy MCSs when combined with bagging. Nevertheless, the results obtained from the other feature selection approaches were not much worse.

2) Benchmarking against the single FURIA-based fuzzy classifier: in our second analysis, we are comparing the FURIA-based fuzzy MCSs derived by the best previous feature selection approach combined with bagging against the single FURIA-based fuzzy classifier. FURIA-based fuzzy MCSs considering bagging and Random-greedy feature selection outperform the single classifier in 8 out of 10 cases for ensembles composed of 3 classifiers, whereas considering ensembles composed of 5 classifiers they obtain better results in 9 out of 10 cases. In addition, FURIA-based fuzzy MCSs considering bagging and Random-greedy feature selection outperform the single classifier in 9 out of 10 cases (+1 tie) for ensembles composed of 7 classifiers. Finally, considering the ensemble size of 10, FURIA-based fuzzy MCSs considering bagging and Random-greedy feature selection obtain better results in all of the cases. Pendigits (2 times + 1 tie) and letter (1 time) are the only datasets for which the single fuzzy classifier outperforms the fuzzy MCSs, probably in due to the large amount of important features contained in the datasets.

TABLE V

FURIA-BASED FUZZY MCSS WITH BAGGING AND RANDOM-GREEDY FEATURE SELECTION.

FURIA - Random-greedy feature selection										
3 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.772	0.016	0.155	0.094	0.041	0.054	0.114	0.446	0.015	0.006
test err.	0.797	0.043	0.368	0.200	0.152	0.128	0.141	0.666	0.031	0.036
5 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.770	0.016	0.132	0.084	0.044	0.037	0.114	0.423	0.014	0.004
test err.	0.796	0.044	0.348	0.198	0.147	0.108	0.139	0.656	0.030	0.030
7 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.761	0.014	0.122	0.079	0.040	0.031	0.113	0.421	0.014	0.004
test err.	0.789	0.042	0.344	0.179	0.146	0.102	0.138	0.646	0.030	0.027
10 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.757	0.014	0.106	0.077	0.035	0.026	0.115	0.410	0.014	0.003
test err.	0.787	0.043	0.334	0.187	0.145	0.096	0.139	0.640	0.030	0.026

TABLE VI

FURIA-BASED FUZZY MCSS WITH BAGGING AND RANDOM SUBSPACE FEATURE SELECTION.

FURIA - Random subspace feature selection										
3 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.772	0.017	0.139	0.080	0.043	0.102	0.120	0.444	0.016	0.005
test err.	0.804	0.043	0.375	0.202	0.165	0.202	0.145	0.665	0.033	0.031
5 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.760	0.017	0.099	0.073	0.041	0.047	0.115	0.417	0.015	0.003
test err.	0.792	0.040	0.339	0.199	0.158	0.132	0.139	0.656	0.031	0.022
7 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.759	0.015	0.087	0.066	0.035	0.034	0.114	0.420	0.014	0.002
test err.	0.793	0.040	0.318	0.195	0.157	0.116	0.139	0.644	0.030	0.018
10 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.754	0.015	0.075	0.062	0.026	0.025	0.117	0.410	0.015	0.002
test err.	0.786	0.041	0.319	0.191	0.147	0.103	0.140	0.638	0.030	0.015

Overall, FURIA-based fuzzy MCSs generated from Bagging and Random-greedy feature selection outperform the single classifier in 36 out of 40 cases (+1 tie).

VI. CONCLUSIONS AND FUTURE WORKS

In this first study, we proposed a methodology in which a bagging approach together with a feature selection technique is used to train FURIA-based fuzzy classifiers in order to obtain a fuzzy rule-based MCS. We used a single winner-based

TABLE VII

RESULTS FOR EACH OF THE FEATURE SELECTION APPROACHES FOR FURIA-BASED FUZZY MCSS GENERATED WITH BAGGING AND FEATURE SELECTION IN THE FORM OF A SUMMARIZED MATRIX.

	Greedy			Random-greedy			Random		
Feature subset size	W	T	L	W	T	L	W	T	L
3	6	1	3	1	2	7	1	1	8
5	3	1	6	3	1	6	3	1	6
7	2	0	8	5	0	5	3	0	7
10	1	1	8	4	1	5	4	1	5
Overall	12	3	25	13	4	23	11	3	26

method on top of the base classifiers. We tested FURIA-based fuzzy MCSs with bagging and feature selection. By using the abovementioned techniques, we aimed to obtain fuzzy MCSs dealing with high dimensional data.

We have conducted experiments over 10 datasets taken from the UCI machine learning repository. We showed that the obtained results are promising and provide a performance advantage over the single FURIA classifier.

One of the next steps we will take in the short future is the measure of the performance of the bagging and the feature selection components separately and testing different feature subset sizes for the feature selection algorithms. Moreover, we could check what evolutionary multiobjective optimization algorithms can bring for the final classifier selection in the view of getting an optimal size of the ensemble. This MCS design approach, called overproduce-and-choose strategy (OCS) [49], [50] is based on the generation of a large number of component classifiers and of the subsequent selection of the subset of them best cooperating. By doing so, the performance of FURIA-based fuzzy MCSs could be improved, while decreasing the number of classifiers in the ensemble, thus obtaining different trade-offs between accuracy and complexity. The other extension to follow is to study other fuzzy reasoning methods to combine the results of the individual members of the ensemble, trying to combine classifiers in a dynamic manner, in a way that a classifier or a set of them is responsible just for a particular data region.

ACKNOWLEDGMENT

This work has been partially supported by the Spanish Ministry of Science and Innovation (MICINN) under project TIN2009-07727, including EDRF fundings.

REFERENCES

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

[2] P. P. Bonissone, J. M. Cadenas, M. C. Garrido, and R. A. Díaz-Valladares. A fuzzy random forest. *International Journal of Approximate Reasoning*, 51(7):729–747, 2010.

[3] 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 (UK), 2007.

[4] C. Marsala. Data mining with ensembles of fuzzy decision trees. In *IEEE Symposium on Computational Intelligence and Data Mining*, pages 348–354, Nashville (USA), 2009.

[5] Y. Nojima and H. Ishibuchi. Genetic rule selection with a multi-classifier coding scheme for ensemble classifier design. *International Journal of Hybrid Intelligent Systems*, 4(3):157–169, 2007.

[6] W. Pedrycz and K.C. Kwak. Boosting of granular models. *Fuzzy Sets and Systems*, 157(22):2934–2953, 2006.

[7] H. Takahashi and H. Honda. A new reliable cancer diagnosis method using boosted fuzzy classifier with a SWEEP operator method. *Journal of Chemical Engineering of Japan*, 38(9):763–773, 2005.

[8] 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.

[9] J. C. Hühn and E. Hüllermeier. FURIA: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, 19(3):293–319, 2009.

[10] J. C. Hühn and E. Hüllermeier. An analysis of the FURIA algorithm for fuzzy rule induction. In *Advances in Machine Learning I*, pages 321–344. 2010.

[11] H. Ishibuchi, T. Nakashima, and M. Nii. *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining (Advanced Information Processing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2004.

[12] W. W. Cohen. Fast effective rule induction. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

[13] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[14] A. Tsymbal, M. Pechenizkiy, and P. Cunningham. Diversity in search strategies for ensemble feature selection. *Information Fusion*, 6(1):83–98, 2005.

[15] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[16] R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[17] J. Gentle, W. Härdle, and Y. Mori. Handbook of computational statistics: Concepts and methods. *Metrika*, 67(2):245–246, March 2008.

[18] T. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[19] P. Panov and S. Džeroski. Combining bagging and random subspaces to create better ensembles. In *IDA'07: Proceedings of the 7th international conference on Intelligent data analysis*, pages 118–129, Berlin, Heidelberg, 2007. Springer-Verlag.

[20] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994.

[21] T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[22] R.E. Banfield, L.O. Hall, K.W. Bowyer, and W.P. Kegelmeyer. A comparison of decision tree ensemble creation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):173–180, 2007.

[23] T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.

[24] D. Optiz and R. Maclin. Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

[25] Z.H. Zhou. Ensembling local learners through multimodal perturbation. *IEEE Transactions of Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(4):725–735, 2005.

[26] L. Xu, A. Krzyzak, and C.Y. Suen. Methods of combining multiple classifiers and their application to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):418–435, 1992.

[27] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[28] W. Pedrycz, A. Breuer, and N.J. Pizzi. Fuzzy adaptive logic networks as hybrid models of quantitative software engineering. *Intelligent Automation and Soft Computing*, 12(2):189–209, 2006.

[29] R. Scherer. Boosting ensemble of relational neuro-fuzzy systems. *Lecture Notes in Computer Science*, 4029:306–313, 2006.

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

[31] F. Hoffmann. Combining boosting and evolutionary algorithms for learning of fuzzy classification rules. *Fuzzy Sets and Systems*, 141(1):47–58, 2004.

[32] L. Sánchez and J. Otero. Boosting fuzzy rules in classification problems under single-winner inference. *International Journal of Intelligent Systems*, 22(9):1021–1034, 2007.

[33] 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*, 39(7):767–771, 2006.

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

[35] C. Z. Janikow. Fuzzy decision trees: issues and methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 28(1):1–14, 1998.

[36] 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 *Third International Workshop on Genetic and Evolving Fuzzy Systems (GEFS)*, pages 11–16, Witten-Bommerholz (Germany), 2008.

[37] 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. *International Journal of Hybrid Intelligent Systems*, 7(1):45–64, 2010.

[38] K. Trawiński, A. Quirin, and O. Cordón. Bi-criteria genetic selection of bagging fuzzy rule-based multiclassification systems. In *IFSA World Congress-EUSFLAT Conference (IFSA/EUSFLAT)*, pages 1514–1519, Lisbon (Portugal), 2009.

[39] K. Trawiński, A. Quirin, and O. Cordón. On the combination of accuracy and diversity measures for genetic selection of bagging fuzzy rule-based multiclassification systems. In *International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 121–127, Pisa (Italy), 2009. IEEE Computer Society.

[40] J.J. Aguilera, M. Chica, M.J. del Jesus, and F. Herrera. Niching genetic feature selection algorithms applied to the design of fuzzy rule based classification systems. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1794–1799, London (UK), 2007.

[41] Y. Nojima and H. Ishibuchi. Designing fuzzy ensemble classifiers by evolutionary multiobjective optimization with an entropy-based diversity criterion. In *International Conference on Hybrid Intelligent Systems and Conference on Neuro-Computing and Evolving Intelligence*, Auckland, New Zealand, 2006.

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

[43] O. Cordón, M.J. del Jesus, and F. Herrera. A proposal on reasoning methods in fuzzy rule-based classification systems. *International Journal of Approximate Reasoning*, 20:21–45, 1999.

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

[45] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illlinois Press, 1949.

[46] P.P. Bonissone, J.M. Cadenas, M.C. Garrido, and R.A. Díaz-Valladares. A fuzzy random forest: Fundamental for design and construction. In *Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge- Based Systems (IPMU'08)*, pages 1231–1238, Málaga (Spain), 2008.

[47] T.G. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

[48] 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 *International Conference on Hybrid Intelligent Systems (HIS)*, pages 549–554, Barcelona (Spain), 2008.

[49] D. Partridge and W.B. Yates. Engineering multiversion neural-net systems. *Neural Computation*, 8(4):869–893, 1996.

[50] E.M. Dos Santos, R. Sabourin, and P. Maupin. A dynamic overproduce-and-choose strategy for the selection of classifier ensembles. *Pattern Recognition*, 41(10):2993–3009, 2008.