On Applying Random Oracles to Fuzzy Rule-Based Classifier Ensembles for High Complexity Datasets

Krzysztof Trawiński[1] **Oscar Cordón**[1,2] **Arnaud Quirin**[3]

[1]European Centre for Soft Computing, 33600 Mieres, Spain
Email: {krzysztof.trawinski, oscar.cordon}@softcomputing.es
[2]Dept. of Computer Science and Artificial Intelligence (DECSAI) and
Research Center on Information and Communication Technologies (CITIC-UGR),
University of Granada, 18071 Granada, Spain
Email: ocordon@decsai.ugr.es
[3]Galician Research and Development Center in Advanced Telecommunications (GRADIANT),
Communications Area, Edif. CITEXVI, local 14, University of Vigo, 36310 Vigo, Spain
Email: aquirin@gradiant.org

Abstract

Fuzzy rule-based systems suffer from the so-called curse of dimensionality when applied to high complexity datasets, which consist of a large number of variables and/or examples. Fuzzy rule-based classifier ensembles have shown to be a good approach to deal with this kind of problems. In this contribution, we would like to take one step forward and extend this approach with two variants of random oracles with the aim that this classical method induces more diversity and in this way improves the performance of the system. We will conduct exhaustive experiments considering 29 UCI and KEEL datasets with high complexity (considering both a number of attributes as well as a number of examples). The results obtained are promising and show that random oracles fuzzy rule-based ensembles can be competitive with random oracles ensembles using state-of-the-art base classifiers in terms of accuracy, when dealing with high complexity datasets.

Keywords: Fuzzy rule-based classifier ensembles, random oracles, bagging, classifier fusion, classifier selection, high complexity datasets

1. Introduction

Fuzzy rule-based classification systems (FRBCSs) are well-known soft computing tools [1, 2], as they are able to model complex, non-linear classification problems via soft boundaries obtained through the fuzzy rules as well as they have capability of knowledge extraction and representation in a way that they could be understood by a human being [3, 4]. FRBCSs, however, have one significant drawback. The main difficulty appears when it comes to deal with a dataset consisting of a high number of variables and/or examples. In such a case FRBCSs suffer from the so-called *curse of dimensionality* [2]. It occurs due to the exponential increase of the number of rules and the number of antecedents within a rule with the growth of the number of inputs of the FRBCS. This issue also causes a problem of scalability in terms of the run time and the memory consumption.

Fuzzy rule-based classifier ensembles (FRBCEs) proved to be a good solution to deal with complex and high dimensional classification problems [5]. In that work, we proposed a methodology for component fuzzy classifier generation. To generate FRBCEs we embedded Fuzzy Unordered Rule Induction Algorithm (FURIA) [6] [1] into a classifier ensemble (CE) framework based on classical CE design approaches such as bagging [7], random subspace [8], and mutual information-based feature selection [9]. The experiments performed showed that out of the three following CE methodologies, that is bagging, feature selection, and bagging with feature selection, the former obtained the best performance when combined with FURIA-based FRBCSs.

We would like to take one step forward and improve the performance of the FRBCEs proposed in [5], when dealing with high complexity datasets. For that purpose, we will incorporate a fast and generic CE technique, namely Random Oracles (ROs) [10, 11], into the CE framework already proposed in order to obtain highly accurate and robust FRBCEs.

ROs is a classical ensemble approach achieving good performance while having several interesting features (a comprehensive study is presented in [10, 11]). This miniensemble replacing the component base classifier is composed of a pair of subclassifiers with a random oracle (a random function, e.g. a random hyperplane) choosing between two

[1]This particular FRBCS is based on scatter fuzzy partitions (instead of the strong fuzzy partitions often in a linguistic form, as commonly used), which allows both to obtain high accuracy and to cope with high dimensional problems.

of them, when an instance is presented in the input. During the training phase, the random oracle splits a dataset into two and feeds each subclassifier with the data from each half-space, while, during the classification phase, it decides which subclassifier makes the final decision to be further used at the ensemble level.

In this contribution, we aim to improve the performance of the FRBCEs proposed in [5] by combining two variants of ROs, a classical ensemble approach, with those FURIA-based fuzzy CEs. The use of ROs has been shown to be able to improve the performance of the CE generation methodologies [10, 11]. In particular, the combination with bagging [7] or random subspace [8] approaches has been apparently emphasized to be the most profitable to the use of ROs, in comparison with the combination with the other approaches. We will show that this FRBCE can not only properly deal with high complexity datasets, but its performance is also competitive with the state-of-the-art RO-based CEs using classical machine learning algorithms such as C4.5 [12] and Naïve Bayes (NB) [13] as base classifiers proposed in [10, 11]. For this purpose, a comprehensive study will be conducted considering 29 high complexity datasets (with either a high number of features or a high number of examples) from the UCI machine learning [2] and from the KEEL dataset [3] repositories to test the accuracy and complexity [4] of the derived CEs. The proposed RO-based FRBCE will be compared with the above-mentioned state-of-the-art RO-based CEs.

The rest of the paper is organized as follows. In the next section, the preliminaries required to introduce our work are reviewed. Section 3 briefly describes the ROs and their incorporation to our FRBCE methodology. The experiments carried out and their analysis are shown in Section 4. Finally, Section 5 concludes this contribution, suggesting also some future research lines.

2. Preliminaries

2.1. Classifier ensemble generation and combination methods

Classically, two kinds of CE design stages are distinguished, each one operating at the different level of the CE structure [14]. The first one is related to the CE generation methodology, which deals with the learning of base classifiers. It aims at generating a set of diverse classifiers, which jointly obtains a high accuracy. Several approaches have been proposed to achieve these objectives along the last decades. The most popular among them are probably data re-sampling techniques. Bagging [7] and boosting [15] are the two leading methods within this approach. In contrast, another group consists of a set of methods inducing the individual classifier diversity through some alternative, specific mechanisms [16] such as feature selection [8] or diversity measures [17, 18]. Hybrid approaches, combining both groups have also been proposed. The most indicative approach could be random forests [19].

The second design task focuses on the combination of the individual decisions provided by the base classifiers to compute the final output of the CE. The two most common approaches are *classifier fusion* and *classifier selection* [20]. The former one is based on the assumption that all the classifiers are trained over the entire feature space and that all ensemble members make independent errors. In contrast, the latter relies on the fact that each classifier is specialized in some local part of the feature space. The weighted majority voting is probably the most extended fusion-based combination method [21]. However, several well known alternatives have been proposed in the literature, including simple functions (majority voting, sum, product, maximum, and minimum) [22] as well as some more advanced techniques [22, 23, 24]. The *classifier selection* strategy consists of either locally selecting the most appropriate (e.g. the best performing) classifier to provide a class label for a given specific example (performing static [25] or dynamic classifier selection [26, 27]) or doing it in a global way by selecting a subset of classifiers that will be used for the entire dataset (e.g. overproduce-and-choose strategy [28]). Hybrid approaches have also been introduced in the literature [25, 29].

One of the most interesting features of ROs is that this approach somehow fits to both families: CE generation methodologies and combination methods (see Sec. 3). It can serve as the base classifier generation strategy as well as it is a hybrid method joining *classifier fusion* and *classifier selection.*

2.2. Bagging fuzzy classifier ensembles

In this contribution, we will follow a methodology for the component fuzzy classifier generation that we previously presented in [5]. To generate FRBCEs we embedded FURIA [6] into a CE framework based on classical CE design approaches [7, 8, 9]. We concluded that pure bagging without additional feature selection obtained the best performance when combined with FURIA-based FRBCSs. Thus, we consider the use of bagging with the entire feature set to generate the initial FURIA-based fuzzy CEs.

In order to build these FRBCEs, a normalized dataset is split into two parts, a training set and a test set. The training set is submitted to an instance selection procedure in order to provide the K individual training sets (the so-called *bags*) to train the K FURIA-based fuzzy FRBCSs. In every case, the bags are generated with the same size as the original training set, as commonly done.

[2]http://archive.ics.uci.edu/ml

[3]http://www.keel.es

[4]Notice that, interpretability issues are not considered in this paper.

The fuzzy classification rules R_j^k considered show a class C_j^k and a certainty degree CF_j^k in the consequent: If x_1^k is A_{j1}^k and ... and x_n^k is A_{jn}^k then Class C_j^k with CF_j^k, $j = 1, 2, \ldots, N$, $k = 1, 2, \ldots, K$. The voting-based fuzzy reasoning method is used to take the decision of the individual subclassifier [30, 31].

After performing the training stage on all the bags in parallel, we get an initial whole FRBCE, which is validated using the training and the test errors as well as a measure of complexity based on the total number of rules in the FRBCSs. The standard majority voting approach is applied as the classifier fusion method [29, 32]: the ensemble class prediction will directly be the most voted class in the component classifiers output set. In the case of a tie, the output class is chosen at random.

3. Using random oracles to design fuzzy rule-based classifier ensembles

A RO [10, 11] is a structured classifier, also defined as a "miniensemble", encapsulating the base classifier of the CE. It is composed of two classifiers and an oracle that decides which one to use. Basically, the oracle is a random function whose objective is to randomly split the dataset into two subsets by dividing the feature space into two regions. Each of the two generated regions and the corresponding data subset is assigned to one classifier. Any shape for the decision surface of the function (in this contribution a hyperplane is considered) can be applied as far as it divides the training set into two subsets at random.

The ROs approach exhibits several interesting features, making it quite unique among the existing CEs solutions:

- It is a generic approach composing a framework in which ROs embed only the base classifier. Thus, it allows a design choice at two different levels: i) any CE strategy can be applied; ii) any classifier learning algorithm can be used. Apart from that, it can be used as the CEs generation method on its own.
- It induces an additional diversity through the randomness coming from the nature of ROs. Generating a set of diverse base classifiers was shown to be fundamental for the overall performance of CEs [17, 33]. Let us emphasize that ROs are applied separately to each of the base classifiers and no training of the oracle is recommended, as it will strongly diminish the desired diversity.
- It embeds the two most common and complementary CE combination methods, i.e. *classifier fusion* and *classifier selection.*
- A wide study has been carried out over several CE generation approaches [10, 11] in order to analyse the influence of ROs on these methods. C4.5 [12] (in [10]) and NB [13] (in [11]) were the base classifiers used. All the CE generation approaches used took advantage of the ROs, outperforming the original CEs in terms of accuracy. Especially, the highest improvement of the accuracy was obtained by random subspace and bagging according to [10].

Two kinds of ROs were presented so far: random linear oracle (RLO) [10, 11] and random spherical oracle (RSO) [11].

3.1. Random linear oracle

RLOs use a randomly generated hyperplane to divide the feature space. To generate RLO the following procedure was proposed:

- Select randomly a pair of examples from the *training set*
- Find the line segment between these points, passing through the middle point M
- Calculate the hyperplane perpendicular to the obtained line segment and containing M

The interested reader is referred to [10, 11] for more details.

3.2. Random spherical oracle

RSO is based on a hypersphere where one classifier is responsible for the subspace inside of that hypersphere, while the second classifier is in charge of the rest of the feature space (outside of the hypersphere). The generation procedure of RSO is as follows [11]:

- Select randomly at least the half ($\geq 50\%$) of the features
- Choose randomly an example from training set to be the center of the hypersphere
- Calculate distances from the center to K examples from training set (chosen at random); the median of these distances is the radius of the hypersphere

Notice that, the random feature subset selection is done in order to improve the randomness, thus the diversity of the RSO. Moreover, the method itself is scalable, meaning that it is weakly affected by the number of attributes and not affected at all by the number of examples.

3.3. Why does it work?

There are no theoretical proofs standing behind the good functionality of the ROs. The existing CE approaches are usually too complex and difficult to analyse. Kuncheva and Rodríguez [10, 11] presented two concepts, which possibly could explain the robustness of the ROs:

1. **High accuracy of the base classifiers.** As the oracle splits the training set into two subsets, each of the two subclassifiers being a component of the RO may have an easier classification task than a single classifier learning over the entire training set. This may lead to higher accuracy obtained by taking the RO as the base classifier.
2. **High diversity of the base classifiers.** Since the oracle is a random function, it induces additional diversity (through its randomness) to the two subclassifiers. Thus, it is quite probable that the set of base classifiers composing CE is more diverse.

3.4. Framework of random oracles fuzzy classifier ensembles

In this subsection, we will detail how the RO-based bagging FRBCEs are designed. To generate RO-based FRBCEs, a normalized dataset is split into two parts, a training set and a test set. The training set is submitted to an instance selection procedure in order to provide K individual training sets (*bags*) to train RO (either RLO or RSO) mini-ensembles composed of the oracle and two Fuzzy Unordered Rule Induction Algorithm (FURIA) [6, 34] subclassifiers. The oracles randomly split the bags into two parts and feed each FURIA classifier with the data from each half-space. As already said, RLO is based on the randomly generated hyperplane, which serves as a mean to divide the feature space. Alternatively, RSO does so using a random hypersphere. In total, $2 \times K$ FURIA-based fuzzy FRBCSs are generated in every case.

Let us emphasize that during the classification phase, the oracle commits an internal classifier selection, that is to say it decides which FURIA subclassifier makes the final decision for the given example to be further used at the ensemble level (classifier fusion).

Of course, we directly use the fuzzy classification rules generated by the FURIA algorithm. These fuzzy rules R_j^k show a class C_j^k and a certainty degree CF_j^k in the consequent: If x_1^k is A_{j1}^k and ... and x_n^k is A_{jn}^k then Class C_j^k with CF_j^k, $j = 1, 2, \dots, J$, $k = 1, 2, \dots, K$, J being a number of rules. The voting-based fuzzy reasoning method is used to take the decision of the individual subclassifier [30, 31].

After the training, we get an initial RO-based bagging FRBCE, which is validated using the training and the test errors, as well as a measure of complexity based on the total number of fuzzy rules obtained from the FURIA classifiers. The standard majority voting approach is applied as the classifier fusion method [29, 32]: the ensemble class prediction will directly be the most voted class in the ROs output set. In the case of a tie the output class is chosen at random.

The global framework of the RO-based bagging FRBCE approach is presented in Fig. 1.

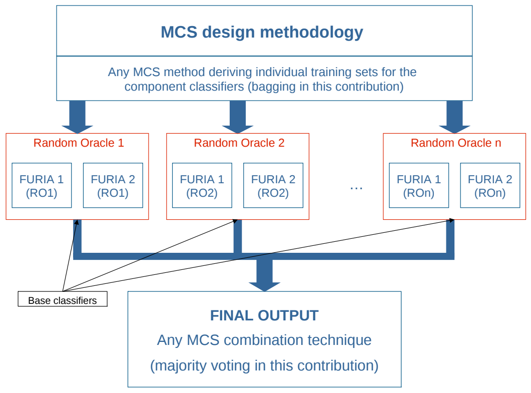
MCS design methodology
Any MCS method deriving individual training sets for the component classifiers (bagging in this contribution)
Random Oracle 1
FURIA 1 (RO1)
FURIA 2 (RO1)
Random Oracle 2
FURIA 1 (RO2)
FURIA 2 (RO2)
...
Random Oracle n
FURIA 1 (ROn)
FURIA 2 (ROn)
Base classifiers
FINAL OUTPUT
Any MCS combination technique
(majority voting in this contribution)

Figure 1: Our framework: after the instance selection, the individual component classifiers are derived by RLO composed of an oracle and two FURIA-based subclassifiers. The final output is taken by means of the majority voting, an inherent feature of bagging.

4. Experiments and analysis of results

This section is devoted to validate our framework using FURIA as a base classifier in RO-based bagging FRBCEs. Firstly, the experimental setup considered is introduced. Then, RLO- and RSO-based bagging FRBCEs are compared with bagging FRBCEs in order to show that ROs have a good influence on the performance of bagging FRBCEs. Furthermore, selected RSO-based bagging FRBCEs are compared with classical RSO-based bagging CEs. By doing so, we want to show that RSO-based bagging FRBCEs are competitive against the state-of-the-art RSO-based bagging CEs using C4.5 [10, 11] and Naïve Bayes [11] as the base classifiers, when dealing with high complexity datasets, thanks to the use of the FURIA algorithm.

4.1. Experimental setup

To evaluate the performance of the RO-based bagging FRBCEs, 29 high dimensional data sets from the UCI machine learning and the KEEL dataset repositories have been selected (see Table 1). Every attribute is tagged as real, integer, or nominal, denoted by “(R/I/N)“ in the table. As it can be seen, the number of features ranges from 7 to 617, while the number of examples does so from 1,941 to 58,000. For illustrative purposes, we show in the table a complexity index computed as follows $\frac{\#ex. \times \#attr.}{10000}$, denoted by "cmpl.".

In order to compare the accuracy of the considered classifiers, we used the Dietterich's 5×2-fold cross-validation (5×2-cv) [35]. The Friedman test and the Iman-Davenport are also used for assessing the statistical significance of the differences between algorithms [36, 37].

Dataset	#ex.	#att.	(R/I/N)	cmpl.	#cl.
abalone	4178	8	(7/0/1)	3.3	28
bioassay_688red	27190	153	(27/126/0)	416.0	2
coil2000	9822	85	(0/85/0)	83.5	2
gas_sensor	13910	128	(128/0/0)	178.0	7
isolet	7797	617	(617/0/0)	481.1	26
letter	20000	16	(0/16/0)	32.0	26
magic	19020	10	(10/0/0)	19.0	2
marketing	6876	13	(0/13/0)	8.9	9
mfeat_fac	2000	216	(0/216/0)	43.2	10
mfeat_fou	2000	76	(76/0/0)	15.2	10
mfeat_kar	2000	64	(64/0/0)	12.8	10
mfeat_zer	2000	47	(47/0/0)	9.4	10
musk2	6598	166	(0/166/0)	109.5	2
optdigits	5620	64	(0/64/0)	36.0	10
pblocks	5474	10	(4/6/0)	5.5	5
pendigits	10992	16	(0/16/0)	17.6	10
ring_norm	7400	20	(20/0/0)	14.8	2
sat	6436	36	(0/36/0)	23.2	6
segment	2310	19	(19/0/0)	4.4	7
sensor_read_24	5456	24	(24/0/0)	13.1	4
shuttle	58000	9	(0/9/0)	52.2	7
spambase	4602	57	(57/0/0)	26.2	2
steel_faults	1941	27	(11/16/0)	5.2	7
texture	5500	40	(40/0/0)	22.0	11
thyroid	7200	21	(6/15/0)	15.1	3
two_norm	7400	20	(20/0/0)	14.8	2
waveform_noise	5000	40	(40/0/0)	20.0	3
waveform1	5000	21	(21/0/0)	10.5	3
wquality_white	4898	11	(11/0/0)	5.4	7

Table 1: Datasets considered.

The Wilcoxon Signed-rank test is used for paired comparisons. The confidence level considered for the null hypothesis rejection of all statistical tests considered is 5%.

4.2. Comparison of RSO-based bagging FRBCEs with RLO-based bagging FRBCEs and bagging FRBCEs

This subsection is devoted to analyze the performance of ROs combined with bagging FRBCEs. We compare them with the bagging FRBCEs approach proposed in [5], a base variant without RO. In order to make a fair comparison, we consider all CEs having a similar complexity based on the total number of rules in the FRBCSs. Notice that, although by embedding ROs into the CE the number of resulting classifiers in the ensemble increases by two (RO includes an oracle and two classifiers for each bag), the total number of rules in the FRBCEs does not necessarily have to increase by the same factor (it will be shown below, when analyzing Table 4). Thus, we consider the generated bagging FRBCEs comprised by 50 classifiers and RO-based bagging FRBCEs comprised by 37 classifiers only to achieve a similar complexity in terms of number of fuzzy rules in both ensembles.

The obtained results over the 29 selected datasets are presented in Table 2, that collects the test errors for the three tested FRBCEs. The best result for a given dataset is presented in bold font. The average "Avg." and standard deviation "Std. Dev." values over the 29 datasets are reported at the bottom of the table.

In view of this table, it can be noticed that both RO-based bagging FRBCEs outperform the original bagging FRBCEs considering the overall average test error. Taking each individual dataset into account, RLO-based bagging FRBCEs outperform bagging FRBCEs in 20 out of 29 cases (+1 tie), while RSO-based bagging FRBCEs do so in another 20 out of 29 cases (+2 ties).

It seems that RSO-based bagging FRBCEs is the approach worth pointing out, as it obtains the lowest overall average test error, as well as the highest number of the best individual results (13+2 ties). Nonetheless, a clear conclusion cannot be drawn as RLO-based bagging FRBCEs are not much inferior in terms of overall average test error.

	BAG	BAG+RLO	BAG+RSO
Dataset	Test err.	Test err.	Test err.
abalone	0.7460	**0.7450**	0.7472
bioassay_688red	**0.0090**	**0.0090**	**0.0090**
coil2000	**0.0601**	0.0602	**0.0601**
gas_sensor	0.0091	0.0082	**0.0081**
isolet	0.0790	**0.0717**	0.0727
letter	0.0799	0.0761	**0.0760**
magic	0.1346	0.1322	**0.1304**
marketing	0.6764	**0.6686**	0.6690
mfeat_fac	0.0549	0.0475	**0.0461**
mfeat_fou	0.1993	0.1969	**0.1924**
mfeat_kar	0.0829	**0.0728**	0.0737
mfeat_zer	0.2221	**0.2193**	0.2220
musk2	0.0351	0.0329	**0.0321**
optdigits	0.0329	**0.0287**	0.0289
pblocks	**0.0288**	0.0350	0.0341
pendigits	0.0160	0.0142	**0.0136**
ring_norm	0.0438	0.0442	**0.0326**
sat	0.1022	0.1011	**0.1007**
segment	0.0333	0.0307	**0.0296**
sensor_read_24	**0.0221**	0.0228	0.0231
shuttle	**0.0008**	0.0009	0.0009
spambase	**0.0579**	0.0643	0.0640
steel_faults	**0.2376**	0.2389	0.2379
texture	0.0305	0.0291	**0.0280**
thyroid	**0.0216**	0.0218	0.0218
two_norm	0.0312	**0.0277**	0.0288
waveform	0.1492	**0.1474**	0.1482
waveform1	0.1484	0.1466	**0.1459**
wquality_white	0.3935	0.3852	**0.3825**
Avg.	0.1289	0.1269	**0.1262**
Std. Dev.	0.1836	0.1826	0.1830

Table 2: A comparison of RO-based bagging FRBCEs (37 classifiers) with bagging FRBCEs (50 classifiers) in terms of accuracy.

These conclusions are confirmed in Table 3, which presents the p-values of the Wilcoxon signed-rank tests between the three FRBCE design approaches (the results showing a significant difference are presented in bold font). Both RO-based bagging FRBCEs show significant differences in comparison to bagging FRBCEs. However, the statistical test did not indicate significant differences between both RO-based bagging FRBCEs.

Comparison	p-value
BAG+RLO vs BAG	**+(0.0024)**
BAG+RSO vs BAG	**+(0.0022)**
BAG+RLO vs BAG+RSO	=(0.1741)

Table 3: Wilcoxon Signed-rank test for the comparison of RO-based bagging FRBCEs with bagging FRBCEs.

As already mentioned, we use the overall number of rules in the FRBCEs as a measure of the ensemble complexity, while the number of classifiers composing the ensemble was fixed after a preliminary study. Table 4 shows the corresponding values for each of the three FRBCEs.

In the light of this table, it can clearly be noticed that both RO-based bagging FRBCEs obtain a lower complexity than the original bagging FRBCEs in terms of overall average number of rules. RLO-based bagging FRBCEs obtain the lowest overall average number of rules, as well as the lowest individual number of rules in 25 out of 29 cases (even though RSO-based bagging FRBCEs are not much inferior). Notice that, the overall standard deviation values are very high due to the large number of rules obtained for the letter dataset. Because of that, we also present the overall average number of rules and the overall standard deviation for the 28 remaining datasets at the bottom of this table.

Dataset	BAG # Rules	BAG+RLO # Rules	BAG+RSO # Rules
abalone	**3990.9**	4298.9	4611.3
bioassay_688red	2754.2	**2311.0**	2374.6
coil2000	2139.7	**1843.0**	1977.6
gas_sensor	4311.8	**3488.9**	3608.0
isolet	6107.6	**5192.9**	5343.9
letter	23533.2	**19452.2**	20189.3
magic	**3881.2**	6381.3	7142.6
marketing	**3198.5**	3593.9	3663.4
mfeat_fac	1736.4	**1502.2**	1537.7
mfeat_fou	2741.4	**2320.2**	2410.6
mfeat_kar	2473.4	**2193.4**	2255.4
mfeat_zer	2504.4	**2147.4**	2246.6
musk2	2163.1	**1763.8**	1770.1
optdigits	3584.6	**3137.7**	3230.0
pblocks	**1329.4**	1431.3	1397.5
pendigits	4395.3	**3628.2**	3701.1
ring_norm	3658.1	3069.8	**2954.5**
sat	4207.2	**3429.5**	3514.7
segment	1175.3	**1084.1**	1169.8
sensor_read_24	1704.8	**1651.9**	1688.2
shuttle	914.3	**849.7**	853.2
spambase	2220.9	**1617.7**	2074.3
steel_faults	2750.4	**2368.9**	2404.4
texture	2912.2	**2621.0**	2725.9
thyroid	1656.5	**1405.4**	1458.1
two_norm	3078.3	**2449.3**	2616.6
waveform	3484.3	**3315.7**	3381.7
waveform1	4152.5	**3457.7**	3503.1
wquality_white	6734.3	**6015.9**	6217.6
Avg.	3775.7	**3380.1**	3518.0
Std. Dev.	4032.7	3382.3	3525.1
Avg. (No letter)	3070.0	**2806.1**	2922.6
Dev. (No letter)	1375.0	1398.0	1491.6

Table 4: A comparison of RO-based bagging FRBCEs (37 classifiers) with bagging CEs (50 classifiers) in terms of complexity (number of rules).

All in all, we may conclude that RO-based bagging FRBCEs significantly outperform bagging FRBCEs both in terms of accuracy and complexity. A decision whether to choose RLO or RSO is not straightforward since RLO obtains slightly lower accuracy, but also lower complexity, while RSO does the opposite (slightly higher accuracy at the cost of a slightly higher complexity). For the purpose of this contribution, which focuses on obtaining highly accurate FRBCEs, we will choose the RSO approach for the further comparisons.

4.3. Comparison of RSO-based FRBCEs with other RSO-based CEs

In this subsection we compare RSO-based bagging FRBCEs with classical RSO-based bagging CEs using C4.5 [10, 11] and Naïve Bayes (NB) [11] as the base classifiers.

In this case, a comparison using the number of rules in the base classifiers used as the complexity measure is rather impossible, as NB is not a rule-based classifier and C4.5 considers tree-based rules.

Table 5 presents the test results achieved by RSO-based bagging FRBCEs and RSO-based bagging CE using C4.5 and NB over the 29 datasets.

In the light of this table, it can be noticed that RSO-based bagging FRBCEs outperform the other approaches considering the overall average test error. It also obtains the highest number of wins (16+2 ties) in the individual datasets. In opposite, RSO-based bagging CEs based on NB turns out to be the worst choice both considering overall average test error and the number of individual best results.

Dataset	FURIA Test err.	C4.5 Test err.	NB Test err.
abalone	**0.7472**	0.7696	0.7624
bioassay_688red	**0.0090**	**0.0090**	0.0153
coil2000	**0.0601**	0.0616	0.1820
gas_sensor	**0.0081**	0.0094	0.3003
isolet	**0.0727**	0.0813	0.1253
letter	0.0760	**0.0658**	0.2926
magic	0.1304	**0.1268**	0.2366
marketing	**0.6690**	0.6745	0.6875
mfeat_fac	**0.0461**	0.0501	0.0655
mfeat_fou	**0.1924**	0.1948	0.2205
mfeat_kar	0.0737	0.0867	**0.0597**
mfeat_zer	**0.2220**	0.2294	0.2473
musk2	0.0321	**0.0283**	0.1121
optdigits	**0.0289**	0.0297	0.0717
pblocks	0.0341	**0.0330**	0.0705
pendigits	**0.0136**	0.0161	0.0861
ring_norm	0.0326	0.0397	**0.0202**
sat	0.1007	**0.0967**	0.1731
segment	**0.0296**	0.0326	0.1198
sensor_read_24	**0.0231**	0.0232	0.3703
shuttle	**0.0009**	**0.0009**	0.0157
spambase	**0.0640**	0.0658	0.1777
steel_faults	0.2379	**0.2286**	0.3429
texture	**0.0280**	0.0351	0.1426
thyroid	0.0218	**0.0215**	0.0393
two_norm	0.0288	0.0327	**0.0222**
waveform	**0.1482**	0.1698	0.1672
waveform1	**0.1459**	0.1654	0.1541
wquality_white	0.3825	**0.3737**	0.5216
Avg.	**0.1312**	0.1357	0.2068
Std. Dev.	0.1819	0.1856	0.1892

Table 5: A comparison of RSO-based bagging CEs using FURIA, C4.5 and NB in terms of accuracy.

In Table 6, the ranking of the three RSO-based CEs considered on each dataset is shown. RSO-based FRBCEs appears 18 times in the first position (with 2 ties) and 11 times in the second position. Hence, it never appears in the third position.

The average rankings of each CE obtained through the Friedman test is shown in Table 7. The Iman-Davenport test indicates significant differences between the algorithms, as the p-value is equal to 3.025066×10^{-7}, which is much lower than the assumed α-value 0.05.

Dataset	FURIA	C4.5	NB
abalone	1	3	2
bioassay_688red	1	1	3
coil2000	1	2	3
gas_sensor	1	2	3
isolet	1	2	3
letter	2	1	3
magic	2	1	3
marketing	1	2	3
mfeat_fac	1	2	3
mfeat_fou	1	2	3
mfeat_kar	2	3	1
mfeat_zer	1	2	3
musk2	2	1	3
optdigits	1	2	3
pblocks	2	1	3
pendigits	1	2	3
ring_norm	2	3	1
sat	2	1	3
segment	1	2	3
sensor_read_24	1	2	3
shuttle	1	1	3
spambase	1	2	3
steel_faults	2	1	3
texture	1	2	3
thyroid	2	1	3
two_norm	2	3	1
waveform	1	3	2
waveform1	1	3	2
wquality_white	2	1	3

Table 6: Performance ranking of the RSO-based CEs using different base classifiers in terms of accuracy.

Algorithm	Ranking
FURIA	1.414
C4.5	1.896
NB	2.689

Table 7: Average Rankings of the Friedman's test

The Wilcoxon signed-rank test confirms these conclusions (see Table 8). It reveals significant differences in favor of RSO-based bagging FRBCEs when comparing with RSO-based bagging CEs using NB. That is not the case when comparing with RSO-based bagging CEs using C4.5, however notice that p-value is equal to 0.0561, which is actually in the border of the confidence level α-value assumed (assuming α-value equal to 0.10 the statistical test would show significant differences).

Comparison	p-value
FURIA vs C4.5	=(0.0561)
FURIA vs NB	+**(8.00e-006)**

Table 8: Wilcoxon Signed-rank test for the comparison of RSO-based bagging FRBCEs (using FURIA) with RSO-based CEs using C4.5 and Naïve Bayes.

To get a deep insight of the results, the dispersion of the results is shown in Fig. 2 by means of boxplots. In this case, the performance advantage of RSO-based bagging FRBCEs over the other approaches is also clear in view of the lowest values obtained (lower quartile, median, and upper quartile). Both classical RSO-based bagging CEs (especially the approach with NB) turn out to be inferior approaches.

Concluding, RSO-based bagging FRBCEs outperform classical RSO-based bagging CEs using C4.5 and NB. Thus, we may draw the conclusion that RO-based bagging FRBCEs successfully deal with high complexity datasets, being competitive with classical RO-based bagging CEs using two standard machine learning base classifiers.

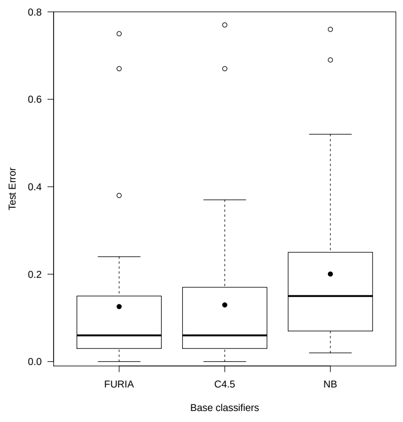

Figure 2: Dispersion of the accuracy of RSO-based bagging FRBCEs, RSO-based bagging CEs using C4.5 and Naïve Bayes.

5. Conclusions and future works

We wanted to demonstrate that random oracles is a CE approach exhibiting several interesting characteristics and when combined with fuzzy classifier ensembles is able to improve its performance. The proposed approach was not only able to deal with high complexity problems, but also obtained a high performance, being competitive with standard base classifiers such as C4.5 and NB in terms of accuracy and complexity (C4.5 only). To show that, we carried out exhaustive experiments using 29 high complexity datasets from the UCI and the KEEL repositories.

These promising conclusions lead to several research lines to follow as future works. Definitely, combining random oracles with other classifier fusion or classifier selection approaches is a challenging objective. Among them, we would like to consider decision templates (classifier fusion) [29], overproduce-and-choose strategy (classifier selection) [27, 28], and dynamic classifier selection (classifier selection) [26, 27].

References

[1] L. I. Kuncheva. *Fuzzy Classifier Design.* Springer, 2000.

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

[3] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena. *Interpretability Issues in Fuzzy Modeling.* Springer Verlag, Berlin Heidelberg, 2003.

[4] J. M. Alonso, L. Magdalena, and G. González-Rodríguez. Looking for a good fuzzy system interpretability index: An experimental approach. *Int. J. Approx. Reason.*, 51:115–134, 2009.

[5] K. Trawiński, O. Cordón, and A. Quirin. On designing fuzzy rule-based multiclassification systems by combining FURIA with bagging and feature selection. *Int. J. Uncert. Fuzz. Knowl.-Based Syst.*, 19(4):589–633, 2011.

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

[7] L. Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, 1996.

[8] T. Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):832–844, 1998.

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

[10] L. I. Kuncheva and J. J. Rodríguez. Classifier ensembles with a random linear oracle. *IEEE Trans. Knowl. Data Eng.*, 19(4):500–508, 2007.

[11] J. J. Rodríguez and L. I. Kuncheva. Naïve bayes ensembles with a random oracle. In *Lect. Notes in Comput. Sci.*, volume 4472, pages 450–458. Springer-Verlag, 2007.

[12] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[13] P. Domingos and M. J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Mach. Learn.*, 29(2-3):103–130, 1997.

[14] B.V. Dasarathy and B.V. Sheela. A composite classifier system design: Concepts and methodology. *Proc. IEEE*, 67(5):708–713, 1979.

[15] R. Schapire. The strength of weak learnability. *Mach. Learn.*, 5(2):197–227, 1990.

[16] Z.H. Zhou. Ensembling local learners through multimodal perturbation. *IEEE Trans. Syst., Man, Cybern. B*, 35(4):725–735, 2005.

[17] L. I. Kuncheva and Ch. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn.*, 51(2):181–207, 2003.

[18] D. Ruta and B. Gabrys. Classifier selection for majority voting. *Inf. Fusion*, 6(1):63–81, 2005.

[19] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.

[20] K. Woods, W.P. Kegelmeyer, and K. Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(4):405–410, 1997.

[21] L. Lam and C.Y. Suen. Application of majority voting to pattern recognition: An analysis of its behavior and performance. *IEEE Trans. Syst., Man, Cybern.*, 27:553–568, 1997.

[22] L. I. Kuncheva. “Fuzzy” versus “nonfuzzy” in combining classifiers designed by boosting. *IEEE Trans. Fuzzy Syst.*, 11(6):729–741, 2003.

[23] A. Verikas, A. Lipnickas, K. Malmqvist, M. Bacauskiene, and A. Gelzinis. Soft combination of neural classifiers: A comparative study. *Pattern Recogn. Lett.*, 20(4):429–444, 1999.

[24] L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin. Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognit.*, 34(2):299–314, 2001.

[25] R. Avnimelech and N. Intrator. Boosted mixture of experts: An ensemble learning scheme. *Neural Comput.*, 11:483–497, 1999.

[26] G. Giacinto and F. Roli. Dynamic classifier selection based on multiple classifier behaviour. *Pattern Recognit.*, 34(9):1879–1881, 2001.

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

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

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

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

[31] O. Cordón, M.J. del Jesus, and F. Herrera. A proposal on reasoning methods in fuzzy rule-based classification systems. *Int. J. Approx. Reason.*, 20:21–45, 1999.

[32] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–238, 1998.

[33] D. Optiz and R. Maclin. Popular ensemble methods: An empirical study. *J. Artif. Intell. Res.*, 11:169–198, 1999.

[34] J. C. Hühn and E. Hüllermeier. An analysis of the FURIA algorithm for fuzzy rule induction. In *in Proc. Adv. Mach. Learn. I*, pages 321–344. 2010.

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

[36] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, 2006.

[37] S. García and F. An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *J. Mach. Learn. Res.*, 9:2677–2694, 2008.