

Towards the Learning from Low Quality Data in a Fuzzy Random Forest ensemble

Jose M. Cadenas

Dpt. Engineering Information and Communications

Faculty of Informatic, University of Murcia

Campus of Espinardo, Murcia, Spain

Email: jcadenas, carmengarrido, raquel.m.e@um.es

M. Carmen Garrido

Raquel Martinez

Piero P. Bonissone

GE Global Research

One Research Circle

Niskayuna, NY 12309, USA

Email: bonissone@ge.com

Abstract—Imperfect information inevitably appears in real situations for a variety of reasons. Although efforts have been made to incorporate imperfect data into classification techniques, there are still many limitations as to the type of data, uncertainty and imprecision that can be handled. In this paper, we will present a Fuzzy Random Forest ensemble for classification and show its ability to handle imperfect data into the learning and the classification phases. Then, we will describe the types of imperfect data it supports. We will devise an augmented ensemble that can operate with others type of imperfect data: crisp, missing, probabilistic uncertainty and imprecise (fuzzy and crisp) values. Additionally, we will perform experiments with datasets used in other papers to show the advantage of being able to express the true nature of imperfect information.

Index Terms—Fuzzy Random Forest; Imperfect Data

I. INTRODUCTION

Classification has always been a challenging problem [1], [2]. Many techniques and algorithms have been developed to address this issue. There is a wide range of classification techniques based on different theoretical proposals [3]–[6]. Unfortunately, most of the conventional techniques did not consider the sources of uncertainty. As a result, incomplete and imprecise data have usually been discarded or ignored in the classifier's learning and in its subsequent classification process. However, this data is inevitable when dealing with real-world applications.

Imperfect information inevitably appears in realistic domains and situations. Instrument errors or corruption from noise during experiments may give rise to information with incomplete data when measuring a specific attribute. In other cases, the extraction of exact information may be excessively costly or unfeasible. Moreover, it might be useful to complement the available data with additional information from an expert, which is usually elicited by imperfect data (interval data, fuzzy concepts, etc). In most real-world problems, data have a certain degree of imprecision. Sometimes, this imprecision is small enough for it to be safely ignored. On other occasions, the imprecision of the data can be modeled by a probability distribution. However, there is a third kind of problem, where the imprecision is significant, and a probability distribution is not the most natural way to model it. This is the case of certain practical problems where the data are inherently fuzzy [7]–[10]. For these cases, it is necessary to incorporate the handling of information with attributes which may present missing and

imprecise values in the classifier's learning and classification phases.

In general, we can say that there are many restrictions to the type of imperfection allowed in the input examples or in the data. For this reason, we propose to extend the treatment of imperfect information with a classification technique based on multiple classifier systems. This technique is a Fuzzy Random Forest (FRF) ensemble. Beside its ability to handle imperfect information, FRF also proved to be effective when dealing with regular data, e.g., without imperfect information [11].

FRF uses a fuzzy decision tree as base classifier. Therefore, FRF leverages the robustness of fuzzy decision trees and tree ensembles, the power of randomness to increase the diversity of the trees in the forest, and the flexibility of fuzzy logic to manage imperfect data. In Section II we review the major elements that constitute the FRF ensemble. In Section III, we describe the types of imperfect data supported by the FRF ensemble and we devise an augmented ensemble that can operate with others type of imperfect data. In Section IV, we perform experiments with imperfect datasets used in other papers to show the advantage of being able to express the true nature of imperfect information. Finally, we present our conclusions in Section V.

II. AN ENSEMBLE BASED ON FUZZY DECISION TREES

FRF was originally presented in [11]. In this section we describe the basic elements that compose a FRF ensemble and the types of data that are supported by this ensemble in both learning and classification phases.

A. Fuzzy Random Forest Learning

FRF learning phase uses Algorithm 1 to generate the FRF ensemble whose trees are fuzzy decision trees. Each tree in the FRF ensemble is a fuzzy tree. Algorithm 2 shows the fuzzy decision tree learning algorithm.

Although the user chooses the size of the FRF, computational studies have shown that good accuracy is obtained on sizes between [100,150] trees, [11]. Algorithm 2 has been designed so that the trees can be constructed without considering all the attributes to split the nodes. Algorithm 2 is an algorithm to construct trees based on ID3, where the numeric attributes have been discretized by a fuzzy partition. The domain of each numeric attribute is represented by trapezoidal fuzzy sets,

Algorithm 1 - FRF ensemble Learning

FRFlearning(*in*: E , *Fuzzy Partition*; *out*: *Fuzzy Random Forest*)
begin

1. Take a random sample of $|E|$ examples with replacement from the dataset E .
2. Apply Algorithm 2 to the subset of examples obtained in the previous step to construct a fuzzy tree, using the fuzzy partition.
3. Repeat steps 1 and 2 until T fuzzy trees are built to constitute the FRF ensemble.

end

Algorithm 2 - Fuzzy Decision Tree Learning

FuzzyDecisionTree(*in*: E , *Fuzzy Partition*; *out*: *Fuzzy Tree*)
begin

1. Start with the examples in E with values $\chi_{Fuzzy_Tree, root}(e) = 1$.
2. Let M be the set of attributes where all numeric attributes are partitioned according to the Fuzzy Partition.
3. Choose an attribute to do the split at the node N .
 - 3.1. Make a random selection of attributes from the set of attributes M .
 - 3.2. Compute the information gain for each selected attribute using the values $\chi_{Fuzzy_Tree, N}(e)$ of each e in node N .
 - 3.3. Choose the attribute such that information gain is maximal.
4. Divide N in children nodes according to possible outputs of the attribute selected in the previous step and remove it from the set M . Let E_n be the dataset of each child node.
5. Repeat steps 3, 4 with each (E_n, M) until the stopping criteria is satisfied.

end

$A_1, \dots, A_f$ so each internal node of the tree, whose division is based on a numerical attribute, generates a child node for each fuzzy set of the partition. Moreover, Algorithm 2 uses a function, denoted by $\chi_{t, N}(e)$, that indicates the degree with which the example e satisfies the conditions that lead to node N of tree t . Each example e is composed of attributes which can take crisp, missing or fuzzy values belonging to the fuzzy partition of the corresponding attribute. Each of these examples (according of the attribute value) has the following treatment:

- Each example e used in the training of the tree t has been assigned an initial value 1 ($\chi_{t, root}(e) = 1$).
- According to the membership degree of the example e to different fuzzy sets of partition of a split based on a numerical attribute:
 - If the value of e for that attribute is crisp, the example e may belong to one or two children nodes ($\mu_{fuzzy_set_partition}(e) > 0$). In this case $\chi_{t, childnode}(e) = \chi_{t, node}(e) \times$

$$\mu_{fuzzy_set_partition}(e).$$

- If the value of e for that attribute is a fuzzy value, it must match one of the sets of the fuzzy partition of the attribute (e will descend to the child node associated with that set). In this case, $\chi_{t, childnode}(e) = \chi_{t, node}(e)$.
- When the example e has a missing value on the attribute used as split in a node, the example e descends to each child node with value $\chi_{t, childnode}(e) = \chi_{t, node}(e) \times \frac{1}{number_outputs_split}$.

The construction process of this fuzzy tree is a modification of fuzzy ID3 algorithm [12] to allow treatment of the attributes with crisp, fuzzy and missing values [11].

B. Fuzzy Random Forest Classification

In this section we describe how the classification is carried out using the FRF ensemble. We present two alternative strategies to obtain the decision of the FRF ensemble for a target example.

When the example e runs through each tree t , we obtain the matrix L_{FRF} of size $(T \times MAX_{N_t})$ with $MAX_{N_t} = \max\{N_1, N_2, \dots, N_T\}$ and N_i the number of reached leaves in tree i . Each element of the matrix is a vector of size I containing the support for every class provided by every activated leaf n on each tree t . Therefore, the matrix L_{FRF} contains all the information generated by the FRF ensemble when it is used to classify an example e and from which it makes its decision or class with certain methods of combination¹. In the Appendix, we present concrete instances of these methods of combination used in the experiments.

The fuzzy classifier module operates on fuzzy trees of the FRF ensemble using one of these two possible strategies:

Strategy 1: Combining the information from the different leaves reached in each tree to obtain the decision of each individual tree and then applying the same or another combination method to generate the global decision of the FRF ensemble. In order to combine the information of the leaves reached in each tree, the $Faggre1_1$ function is used and the $Faggre1_2$ function is used to combine the outputs obtained with $Faggre1_1$ (functions that define the combination method), and so, we obtain vector D_{FRF} (vector with size I that indicates the confidence assigned by the FRF ensemble to each class i).

Strategy 2: Combining the information from all reached leaves from all trees to generate the global decision of the FRF ensemble. We use function $Faggre2$ (function that define the combination method) to combine the information generated by all the leaves obtaining vector D_{FRF} (vector with size I that indicates the confidence assigned by the FRF ensemble to each class i).

From vector D_{FRF} , obtained with Strategy 1 or 2, FRF ensemble assigns class c to example e such that

$$c = \arg \max_{i, i=1, \dots, I} \{D_{FRF_i}\}$$

¹In [11] a set of combination methods are defined.

As we can see, the output of FRF ensemble (regardless of the strategy) is obtained from vector D_FRF , which contains the membership degree of the target example to each class. Finally, the FRF ensemble assigns the class c with maximum value to example e .

III. EXTENDING FRF ENSEMBLE

FRF ensemble is a versatile technique, which is able to adapt itself to new elements. Having introduced the FRF ensemble, we now describe an extension of this technique to allow the incorporation of new imprecise data. This extension involves changes in the learning and classification phases of the technique. In this section, we will describe the new types of imprecise data that we will incorporate and the new elements and changes needed to carry out the FRF ensemble's extension.

A. New types of imprecise data

In Section II-A we described the types of data supported by the FRF ensemble, in both the learning and classification phases, as originally proposed in [11]. Now, we want to extend the information processing in the FRF ensemble to incorporate data containing attributes measured by fuzzy values (which may be different from the fuzzy values which constitute the fuzzy partition of the attribute), and attributes with interval values. Furthermore, we want to allow the class attribute to be set-valued. Figure 1 illustrates the proposed extension.

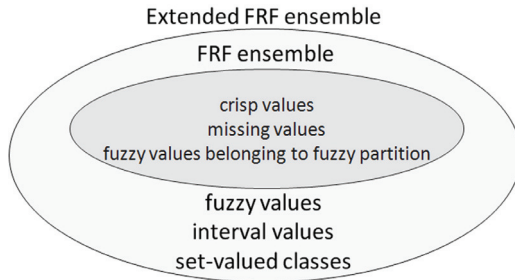


Fig. 1. Extending information processing in FRF

To incorporate these new types of data, we need to modify the following elements of FRF ensemble:

- Define an extended function to measure the membership degree of these new types of data in the fuzzy partitions of the numerical attributes used by the ensemble.
- The definition of this function requires a modification of the *information gain* function in both learning and classification phases of FRF ensemble.
- To incorporate set-valued classes, we need to modify and to define the learning and classification phases.

Let's further analyze these new elements and their required modifications.

B. Fuzzy and Interval values

To incorporate attribute values measured by a fuzzy value (fuzzy set) different from the sets of the partition of the attribute, or attributes measured by interval values, we must extend the function that measures the membership degree of

these types of values to the different fuzzy sets forming the partition of the numeric attributes. In [13] we can find a review of similarity measures. We define the measure $\mu_{simil}(\cdot)$ which captures change in the value $\chi_{t,node}(e)$ of the example e when it descends in the fuzzy tree. e has attributes whose values are expressed by intervals or fuzzy values that do not match any of the fuzzy values of the partition used by the fuzzy tree and therefore by the ensemble. This function μ_{simil} will be used in both learning and classification phases of extended FRF ensemble.

In these cases, we use a membership function ($\mu_{simil}(\cdot)$) that, given the attribute "Attr" to be used to split a node, measures the similarity between the values of the fuzzy partition of the attribute and fuzzy values or intervals of the example in that attribute. When the example e must descend to different child nodes, the function $\chi(\cdot)$ is calculated as $\chi_{t,childnode^i}(e) = \chi_{t,node} \times \mu_{simil}^i(e)$.

Function $\mu_{simil}^i(e)$ is defined as:

$$\mu_{simil}^i(e) = \frac{\int_{\mathbb{R}} (\min\{\mu_e(x), \mu_{A_i}(x)\}) dx}{\sum_{i=1}^f \int_{\mathbb{R}} (\min\{\mu_e(x), \mu_{A_i}(x)\}) dx}, \quad i = 1, \dots, f$$

where:

- $\mu_e(x)$ is the membership function that represents the fuzzy or interval value in the example e
- $\mu_{A_i}(x)$ is the membership function that represents the i -th fuzzy value of the partition of the attribute *Attr*
- f is the cardinality of the partition of the attribute *Attr*

C. Set-valued classes

We assume that if an example is labeled with a set-valued we are not stating that it belongs to all categories at time (which is not an imprecise assert). We are expressing that we are not sure about the class of the example, [14].

As described in Section II, the examples in the dataset are weighted examples (initially with weight 1). When we incorporate in FRF ensemble examples whose class can be expressed by a set of values, it is necessary to modify the way we treat this weight in the learning phase of FRF ensemble. Thus, step 3.2 of Algorithm 2 should calculate the information gain considering the example as belonging to *each* potential class. For this, the example in step 1 of the algorithm, must be replicated for each class of its class attribute, using an associated weight of $1/(n \text{ values classes of example})$. Therefore, the initial weight of the replicated examples will be $1/(n \text{ values classes of example})$.

When the ensemble is learned, we need to extend the classification phase in which the function $\mu_{simil}(e)$ is used.

As an example descends through various branches of the tree (using function $\mu_{simil}(e)$), different leaf nodes are activated. Each of these activated leaf nodes assigns a weight to each possible classes. According to the combination method used, we obtain from the FRF ensemble a weight for each class. The resulting weight vector of classes is denoted by D_FRF .

At this point, when we perform a test of the FRF ensemble results to classify a dataset we can follow the decision process:

```

If class(e)==classFRF(e) and size(class(e))==1 then successes++
else
  If class(e) ∩ classFRF(e) ≠ ∅ then success_or_error++
  else errors++

```

As result of this test, we obtain the interval $[min_error, max_error]$ where min_error is calculated considering only errors indicated in the variable $errors$ from the previous process and max_error is calculated considering as errors $errors + success_or_error$.

D. Changes of learning and classification algorithms

Learning and classification phases should now be modified to allow the FRF ensemble to learn, infer, and classify from an example expressed with the types of values presented in Section III-A. This subsection shows these modification.

Algorithm 2 is modified in steps 1 and 3.2. The new step 1 is “Start with the examples in E with values $\chi_{Fuzzy_Tree, root}(e) = 1/(n \text{ values classes of example})$ ” and the new step 3.2 is “Compute the information gain for each selected attribute using the values $\chi_{Fuzzy_Tree, N}(e)$ of each e in node N taking into account the function $\mu_{simil}(e)$ for the cases required.”

FRF Classification with Strategy 1 is modified. The decisions of trees are modified because obtaining of matrix L_{FRF} takes into account the function μ_{simil} for the cases required. On the other hand, the decision of FRF ensemble is modified due to FRF ensemble assigns the set of classes c to example e such that $c = \arg \max_{i, i=1, \dots, I} \{D_{FRF_i}\}$.

Finally, FRF Classification with Strategy 2 is modified. The decisions of tree are modified because obtaining of matrix L_{FRF} takes into account the function $\mu_{simil}(e)$ for the cases required. Again, FRF ensemble assigns the set of classes c to example e such that $c = \arg \max_{i, i=1, \dots, I} \{D_{FRF_i}\}$.

IV. EXPERIMENTS AND RESULTS

In this section we describe several experiments. These experiments are designed to measure the behavior of FRF ensemble using datasets and results proposed in [14], [15]. We define the experimental settings quite similar to those proposed in [14], [15] to compare with their results.

A. Datasets and parameters for FRF ensemble

To obtain these results we have used real-world datasets about medical diagnosis and high performance athletics [14], [15], whose characteristics are shown in Table I.

TABLE I
DATASETS

Dataset	Abbr	$ E $	$ M $	I
Long-4	Long-4	25	4	2
100ml-4-I	ml-4-I	52	4	2
100ml-4-P	ml-4-P	52	4	2
Dyslexia	Dyslexic	65	12	4

The table shows the number of examples ($|E|$), the number of attributes ($|M|$) and the number of classes (I) for each

dataset. “Abbr” indicates the abbreviation of the dataset used in the experiments.

All FRF ensembles use a forest size of 100 trees. The number of attributes chosen at random at a given node is $\log_2(|\cdot| + 1)$, where $|\cdot|$ is the number of available attributes at that node, and each tree of the FRF ensemble is constructed to the maximum size (node pure or set of available attributes is empty) and without pruning.

In experiments described in [14], [15], fuzzy partitions used are uniform. We will use both uniform and non-uniform fuzzy partitions. These last ones are obtained by the NIP1.5 tool available in “<http://heurimind.inf.um.es/>”.

B. Experimentation with low quality data

These experiments were conducted to test the accuracy of FRF ensemble applied to real-world datasets with imperfect values and compare the results with the ones obtained by the GFS classifier proposed in [14].

In these experiments we used the available datasets in “<http://sci2s.ugr.es/keel/>” and the available results in [14], [15]. These are datasets from two different real-world problems. The first one is related to the composition of teams in high performance athletics and the second one is a medical diagnosis problem.

1) *High performance athletics*: The score of an athletics team is the sum of the individual scores of the athletes in the different events. It is the coach’s responsibility to balance the capabilities of the different athletes in order to maximize the score of a team according to the regulations. The variables that define each problem are as follows:

- There are four indicators for the long jump that are used to predict whether an athlete will pass a given threshold: the ratio between the weight and the height, the maximum speed in the 40 meters race, and the tests of central (abdominal) muscles and lower extremities;
- There are also four indicators for the 100 meters race: the ratio between weight and height, the reaction time, the starting or 20 m speed, and the maximum or 40 m speed.

A more detailed description of this problem may be found in [14]. The datasets used in this experiment are the following:

- Dataset “Long-4”: Dataset used to predict whether an athlete will improve certain threshold in the long jump, given the indicators mentioned before. The set has 25 examples (25 athletes), 4 attributes, 2 classes, and no missing values. All attributes, including the output attribute, are interval-valued.
- Dataset “100ml-4-I”: Used for predicting whether a mark in the 100 meters sprint race is being achieved. Actual measurements are taken by three observers, and are combined into the smallest interval that contains them. The set has 52 examples, 4 attributes, 2 classes, and no missing values. All attributes, including the output attribute, are interval-valued.
- Dataset “100ml-4-P”: Same dataset as “100ml-4-I” but the measurements have been replaced by the subjective grade

TABLE II
RESULTS FOR DATASETS ML-4-I, ML-4-P AND LONG-4 WITH UNIFORM (FIVE LABELS/ATTRIBUTE) AND NON-UNIFORM FUZZY PARTITIONS.

Technique		Dataset					
		ml-4-I		ml-4-P		Long-4	
		Train	Test	Train	Test	Train	Test
Uniform fuzzy partition	FFR _{SM1}	[0.229,0.421]	[0.250,0.443]	[0.075,0.269]	[0.120,0.333]	[0.156,0.436]	[0.217,0.483]
	FRF _{SM2}	[0.226,0.419]	[0.233,0.427]	[0.081,0.274]	[0.100,0.297]	[0.151,0.427]	[0.217,0.483]
	FRF _{MWL1}	[0.209,0.402]	[0.233,0.427]	[0.092,0.287]	[0.120,0.317]	[0.196,0.467]	[0.217,0.483]
	FRF _{MWL2}	[0.207,0.399]	[0.250,0.443]	[0.094,0.286]	[0.120,0.317]	[0.187,0.462]	[0.217,0.483]
	FRF _{MWLT1}	[0.207,0.399]	[0.233,0.427]	[0.094,0.286]	[0.120,0.317]	[0.196,0.475]	[0.217,0.483]
	FRF _{MWLT2}	[0.199,0.391]	[0.250,0.443]	[0.094,0.286]	[0.120,0.317]	[0.191,0.471]	[0.217,0.483]
Non-uniform Fuzzy partition	FFR _{SM1}	[0.139,0.331]	[0.150,0.343]	[0.098,0.291]	[0.133,0.310]	[0.120,0.404]	[0.200,0.467]
	FRF _{SM2}	[0.141,0.333]	[0.150,0.343]	[0.096,0.288]	[0.093,0.290]	[0.115,0.391]	[0.200,0.467]
	FRF _{MWL1}	[0.077,0.269]	[0.093,0.287]	[0.075,0.269]	[0.073,0.270]	[0.116,0.396]	[0.100,0.417]
	FRF _{MWL2}	[0.060,0.252]	[0.093,0.287]	[0.077,0.269]	[0.073,0.270]	[0.102,0.382]	[0.100,0.367]
	FRF _{MWLT1}	[0.077,0.269]	[0.093,0.287]	[0.075,0.267]	[0.073,0.270]	[0.107,0.387]	[0.150,0.417]
	FRF _{MWLT2}	[0.062,0.254]	[0.093,0.287]	[0.077,0.269]	[0.073,0.270]	[0.094,0.373]	[0.067,0.333]
Crisp [14]		0.259	0.384	0.288	0.419	0.327	0.544
GGFS [14]		[0.089,0.346]	[0.189,0.476]	[0.076,0.320]	[0.170,0.406]	[0.000,0.279]	[0.349,0.616]

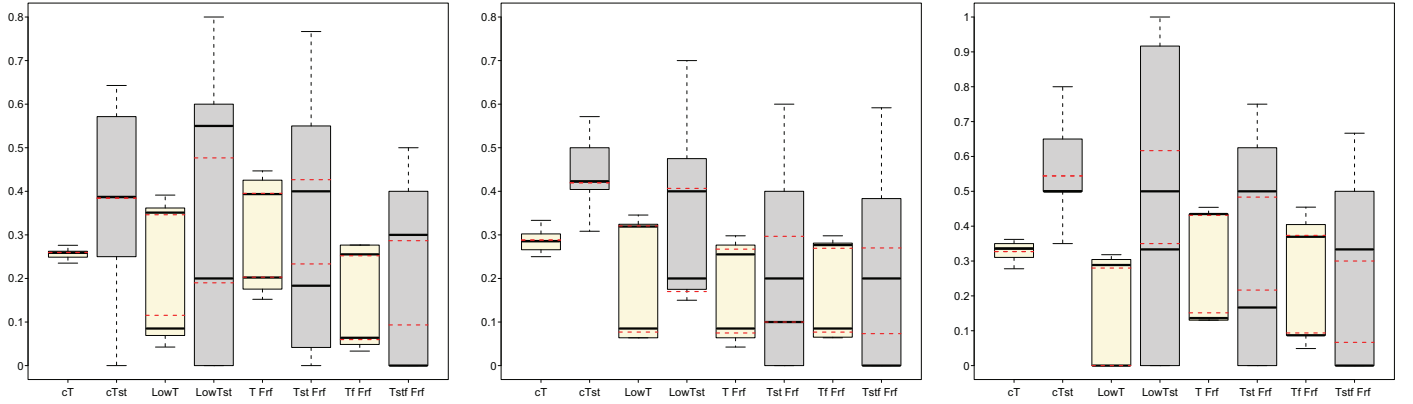


Fig. 2. Boxplots ml-4-I, ml-4-P and Long-4 with five labels/attribute. cT and cTst are the results for train and test respectively of the crisp algorithm proposed in [14]. LowT and LowTst are the results for train and test of the extended algorithm proposed in [14]. T-Frf, Tst-Frf, Tf-Frf and Tstf-Frf are the results for train and test of FRF with uniform and non-uniform fuzzy partitions (results marked in bold).

the trainer has assigned to each indicator (i.e. “reaction time is low” instead of “reaction time is 0.1 sec”).

As in [14], we have used a 10 fold cross-validation design for all datasets. Table II shows the results obtained in [14] and the ones obtained by the FRF ensemble with the six combination methods proposed in Appendix. Except for the crisp algorithm proposed in [14], in this table, the interval $[mean_min_error, mean_max_error]$ obtained for each dataset according to the decision process described in Section III-C, is shown.

In these experiments we used different types of fuzzy partitions of the attributes. We used uniform fuzzy partitions with size of five labels/attribute, as commented in [14]. In addition, we used non-uniform fuzzy partitions. These last ones are obtained by the NIP1.5 tool and they have (3,0,3,3), (0,2,3,6) and (3,3,2,0) labels/attribute for “100ml-4-I”, “100ml-4-P” and “100ml-4-P”, respectively. To obtain these fuzzy partitions for each dataset, attributes with interval values have been changed by its average value.

In Figure 2 is shown the boxplots with the best results of FRF ensemble. Observe that the boxplots of the imprecise experiments are not standard. We have used the extended boxplot proposed in [14]. They propose to use a box showing the 75% percentile of the maximum and the 25% percentile of the minimum error (thus the box displays at least the 50% of data). In addition, they represent the interval-valued median of the maximum and minimum result. For this reason, they draw two marks inside the box.

In Figure 2, we show the results of FRF with uniform and non-uniform fuzzy partitions marked in bold and the results of Crisp [14] and GGFS [14].

The results obtained by the extended GPS proposed in [14] and FRF ensemble, are very promising because we are representing the information in a more natural and appropriate way, and in this problem, we are allowing the collection of knowledge of the coach by ranges of values and linguistic terms. The results of FRF ensemble are very competitive.

TABLE III
COMPARATIVE RESULTS OF FRF WITH UNIFORM AND NON-UNIFORM FUZZY PARTITIONS FOR DATASETS DYSLEXIC-12, DYSLEXIC-12-01 AND DYSLEXIC-12-12.

Technique		Dataset					
		Dyslexic-12		Dyslexic-12-01		Dyslexic-12-12	
		Train	Test	Train	Test	Train	Test
Uniform fuzzy partition	FRF _{SM1} (four labels)	[0.000,0.235]	[0.011,0.441]	[0.000,0.288]	[0.000,0.443]	[0.000,0.265]	[0.000,0.434]
	FRF _{SM2} (four labels)	[0.000,0.231]	[0.014,0.447]	[0.000,0.205]	[0.000,0.412]	[0.000,0.247]	[0.000,0.415]
	FRF _{MWL1} (four labels)	[0.002,0.258]	[0.004,0.436]	[0.000,0.308]	[0.000,0.436]	[0.000,0.290]	[0.000,0.435]
	FRF _{MWL2} (four labels)	[0.001,0.259]	[0.006,0.443]	[0.000,0.312]	[0.000,0.445]	[0.000,0.292]	[0.000,0.436]
	FRF _{MWLT1} (four labels)	[0.002,0.260]	[0.005,0.435]	[0.000,0.279]	[0.000,0.425]	[0.000,0.286]	[0.000,0.432]
	FRF _{MWLT2} (four labels)	[0.001,0.256]	[0.006,0.441]	[0.000,0.286]	[0.000,0.430]	[0.000,0.285]	[0.000,0.432]
Uniform fuzzy partition	FRF _{SM1} (five labels)	[0.000,0.250]	[0.004,0.465]	[0.000,0.307]	[0.000,0.459]	[0.000,0.275]	[0.000,0.443]
	FRF _{SM2} (five labels)	[0.000,0.251]	[0.001,0.515]	[0.000,0.205]	[0.000,0.421]	[0.000,0.264]	[0.000,0.473]
	FRF _{MWL1} (five labels)	[0.003,0.284]	[0.008,0.448]	[0.000,0.339]	[0.000,0.448]	[0.000,0.308]	[0.000,0.451]
	FRF _{MWL2} (five labels)	[0.001,0.278]	[0.004,0.443]	[0.000,0.335]	[0.000,0.449]	[0.000,0.305]	[0.000,0.447]
	FRF _{MWLT1} (five labels)	[0.002,0.280]	[0.007,0.442]	[0.000,0.309]	[0.000,0.431]	[0.000,0.307]	[0.000,0.435]
	FRF _{MWLT2} (five labels)	[0.001,0.277]	[0.004,0.440]	[0.000,0.309]	[0.000,0.435]	[0.000,0.303]	[0.000,0.445]
Non-uniform Fuzzy partition	FRF _{SM1}	[0.000,0.320]	[0.002,0.511]	[0.000,0.282]	[0.000,0.413]	[0.000,0.405]	[0.000,0.477]
	FRF _{SM2}	[0.000,0.327]	[0.001,0.515]	[0.000,0.253]	[0.000,0.389]	[0.000,0.402]	[0.000,0.469]
	FRF _{MWL1}	[0.000,0.261]	[0.003,0.419]	[0.000,0.264]	[0.000,0.400]	[0.000,0.335]	[0.000,0.422]
	FRF _{MWL2}	[0.000,0.270]	[0.003,0.423]	[0.000,0.276]	[0.000,0.407]	[0.000,0.343]	[0.000,0.414]
	FRF _{MWLT1}	[0.000,0.264]	[0.004,0.419]	[0.000,0.243]	[0.000,0.386]	[0.000,0.331]	[0.000,0.422]
	FRF _{MWLT2}	[0.000,0.267]	[0.003,0.417]	[0.000,0.259]	[0.000,0.394]	[0.000,0.343]	[0.000,0.418]
Crisp CF ⁰ (four labels) [15]		0.444	[0.572,0.694]	0.336	[0.452,0.533]	0.390	[0.511,0.664]
GGFS (four labels) [14]		—	[0.421,0.558]	—	[0.219,0.759]	—	[0.199,0.757]
GGFS CF ⁰ (four labels) [15]		[0.003,0.237]	[0.405,0.548]	[0.005,0.193]	[0.330,0.440]	[0.003,0.243]	[0.325,0.509]
Crisp CF ⁰ (five labels) [15]		0.556	[0.614,0.731]	0.460	[0.508,0.605]	0.485	[0.539,0.692]
GGFS (five labels) [14]		—	[0.490,0.609]	—	[0.323,0.797]	—	[0.211,0.700]
GGFS CF ⁰ (five labels) [15]		[0.038,0.233]	[0.480,0.621]	[0.000,0.187]	[0.394,0.522]	[0.000,0.239]	[0.393,0.591]

2) *Diagnosis of dyslexic*: Dyslexia is a learning disability in people with normal intellectual coefficient, and without further physical or psychological problems explaining such disability. A more detailed description of this problem can found in [14], [15].

In these experiments, we have used three different datasets. Their names are “Dyslexic-12”, “Dyslexic-12-01” and “Dyslexic-12-12”. Each dataset has 65 examples, 4 classes and 12 attributes. The output variable for each of these datasets is a subset of the labels that follow: • No dyslexic; • Control and revision; • Dyslexic; and • Inattention, hyperactivity or other problems.

Both input and the output exhibit imprecision and missing values. These three datasets differ only in their outputs:

- “Dyslexic-12” comprises the four mentioned classes.
- “Dyslexic-12-01” does not make use of the class “control and revision”, whose members are included in class “no dyslexic”.
- “Dyslexic-12-12” does not make use of the class “control and revision”, whose members are included in class “dyslexic”.

All experiments are repeated 100 times for bootstrap resamples with replacement of the training set. The test set comprises the “out of the bag” elements. As mentioned above, the boxplots of the imprecise experiments are not standard. We have used the extended boxplot proposed in [14].

In these experiments we used different types of fuzzy

partitions of the attributes. We used uniform fuzzy partitions with size of four and five labels/attribute, as commented in [14], [15]. In addition, we used non-uniform fuzzy partitions. These last ones are obtained by the NIP1.5 tool and they have (0,0,2,0,0,7,4,0,0,0,3,3), (0,2,2,2,3,3,4,0,0,2,0,2) and (0,0,3,2,2,3,0,0,2,2,2,2) labels/attribute for “Dyslexic-12”, “Dyslexic-12-01” and “Dyslexic-12-12P”, respectively. To obtain these fuzzy partitions for each dataset, attributes with interval values have been changed by its average value.

In Table III, we show the results obtained when we run FRF ensemble with these type of partitions for datasets “Dyslexic-12”, “Dyslexic-12-01” and “Dyslexic-12-12”. Also, in Table III, we compare the results obtained by FRF ensemble with the best ones obtained in [15]. Again, in this table, the interval $[mean_min_error, mean_max_error]$ obtained for each dataset according to the decision process described in Section III-C, is shown. The statistical significance of the differences is displayed in Figure 3. In Figure 3, we show the results with FRF with uniform and non-uniform fuzzy partitions marked in bold and the results of Crisp CF⁰ [15], GGFS [14] and GGFS CF⁰ [15] with four labels.

As we can see, FRF with non-uniform fuzzy partitions obtains better results in test that FRF with uniform fuzzy partitions. FRF ensemble is a significant improvement over the crisp GFS. As also discussed in [15] is preferable to use an algorithm which is capable of learning with low quality data than removing the imperfect information and using a

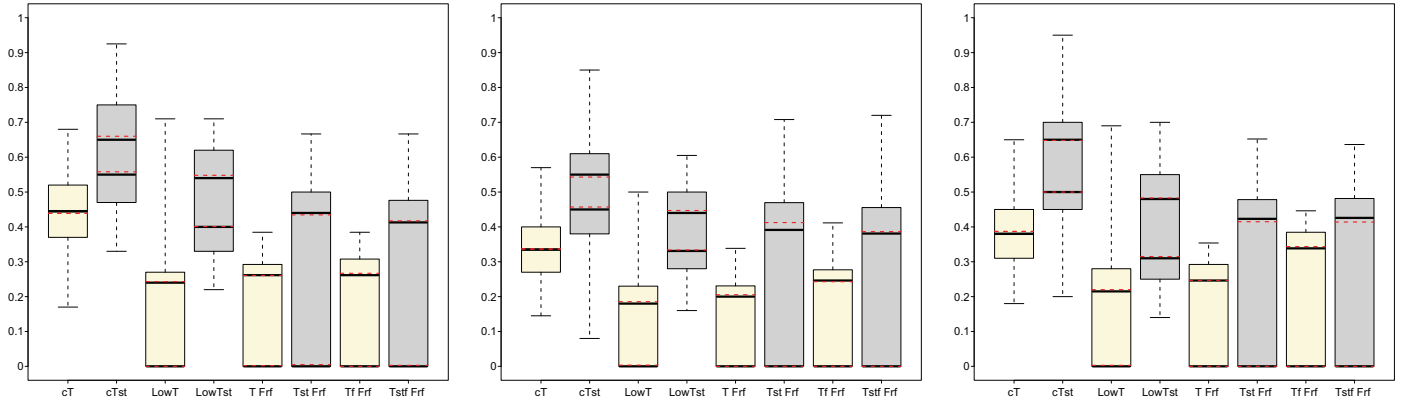


Fig. 3. Boxplots Dyslexic-12, Dyslexic-12-01 and Dyslexic-12-12 with four labels/attributes. cT and cTst are the results for train and test, respectively, of crisp algorithm CF_0 proposed in [15]. LowT and LowTst are the results of the extended algorithm CF_0 proposed in [15]. T-Frf, Tst-Frf, Tf-Frf and Tstf-Frf are the results for train and test of FRF with uniform and non-uniform fuzzy partitions (results marked in bold).

conventional algorithm.

The results of the dataset “Dyslexic-12-01” are better than “Dyslexic-12” and “Dyslexic-12-12”. Moreover, if the class “control and revision” is not used, the individuals in this class tend to be assigned to the group “no dyslexic” rather than to the group “dyslexic”.

V. CONCLUSIONS

We have presented FRF ensemble, which is an ensemble based on fuzzy decision trees. The FRF ensemble can manage imperfect data directly and has a good classification performance with relatively small ensembles sizes. The FRF ensemble has been extended to include the treatment of new types of imperfect data in both input and output attributes. This technique is very versatile due to the ease with which we can extend it to new situations.

We have presented experimental results obtained by applying the FRF ensemble to various datasets. FRF ensemble has been applied to real-problems datasets, such as the composition of teams in high performance athletics and a medical diagnosis problem. The results obtained by FRF ensemble are promising, concluding, as in [14], [15], that by using ranges of values and linguistic terms instead of crisp, imputed numbers, we better capture the nature of the underlying information.

Our conclusion, as many papers in the literature are indicating, is that it is necessary to design and/or adapt classification techniques so they can manipulate real data that can be imperfect in some cases. The transformation of these imperfect values to (imputed) crisp values may cause undesirable effects with respect to accuracy of the technique.

ACKNOWLEDGMENT

Supported by the project TIN2008-06872-C04-03 of the MICINN of Spain and European Fund for Regional Development. Thanks also to the Funding Program for Research Groups of Excellence with code 04552/GERM/06 granted by the “Fundación Séneca”, Murcia, Spain. R. Martínez is supported by the scholarship program FPI from the “Fundación

Séneca” of Spain. Thanks to Luciano Sánchez and Ana Palacios for their help in creating extended boxplot.

APPENDIX

In this appendix, we use the notation above defined in Section II-B. For each strategy for fuzzy classifier module in the FRF ensemble that is described in Section II we can define several functions $Faggre1_1$, $Faggre1_2$ and $Faggre2$. The overall set of functions is described in [11]. In this appendix, the utilized methods in this paper are described in detail:

Non-trainable methods - In these combination methods, a transformation is applied to the matrix L_FRF so that each leaf reached assigns a simple vote to the majority class.

For $t = 1, \dots, T$

For $n = 1, \dots, N$

For $i = 1, \dots, I$

$$L_FRF_{modt,n,i} = \begin{cases} 1 & \text{if } i = \arg \max_{j, j=1,\dots,I} \{L_FRF_{t,n,j}\} \\ 0 & \text{otherwise} \end{cases}$$

Within this group we define the following methods. We get two versions depending on the strategy used:

- *Simple Majority vote:*

Strategy 1 $\rightarrow$ method SM1

The function $Faggre1_1$ is defined as:

$$Faggre1_1(t, i, L_FRF) =$$

$$= \begin{cases} 1 & \text{if } i = \arg \max_{j, j=1,\dots,I} \left\{ \sum_{n=1}^{N_t} L_FRF_{modt,n,j} \right\} \\ 0 & \text{otherwise} \end{cases}$$

In this method, each tree t assigns a simple vote to the most voted class among the N_t reached leaves by example e in the tree. We will denote by T_FRF to the out matrix of $Faggre1_1$.

The function $Faggre1_2$ is defined as:

$$Faggre1_2(i, T_FRF) = \sum_{t=1}^T T_FRF_{t,i}$$

Strategy 2 → method SM2

For Strategy 2 it is necessary to define the function $Faggre2$ combining information from all leaves reached in the ensemble by example e . Thus, the function $Faggre2$ is defined as:

$$Faggre2(i, L_FRF) = \sum_{t=1}^T \sum_{n=1}^{N_t} L_FRF_{modt,n,i}$$

Trainable explicitly dependent methods-

• Majority vote Weighted by Leaf:

In this combination method, a transformation is applied to the matrix L_FRF so that each leaf reached assigns a weighted vote to the majority class. The vote is weighted by the degree of satisfaction with which example e reaches the leaf.

For $t = 1, \dots, T$
 For $n = 1, \dots, N$
 For $i = 1, \dots, I$

$$L_FRF_{modt,n,i} = \begin{cases} \chi_{t,n}(e) & \text{if } i = \arg \max_{j, j=1, \dots, I} \{L_FRF_{t,n,j}\} \\ 0 & \text{otherwise} \end{cases}$$

Again, we have two versions according to the strategy used.

Strategy 1 → method MWL1

The functions $Faggre1_1$ and $Faggre1_2$ are defined as:

$$Faggre1_1(t, i, L_FRF) = \begin{cases} 1 & \text{if } i = \arg \max_{j, j=1, \dots, I} \left\{ \sum_{n=1}^{N_t} L_FRF_{modt,n,j} \right\} \\ 0 & \text{otherwise} \end{cases}$$

$$Faggre1_2(i, T_FRF) = \sum_{t=1}^T T_FRF_{t,i}$$

Strategy 2 → method MWL2

The function $Faggre2$ is defined as:

$$Faggre2(i, L_FRF) = \sum_{t=1}^T \sum_{n=1}^{N_t} L_FRF_{t,n,i}$$

• Majority vote Weighted by Leaf and by Tree:

Again, in this combination method, a transformation is applied to the matrix L_FRF so that each leaf reached assigns a weighted vote to the majority class. The vote is weighted by the degree of satisfaction with which example e reaches the leaf.

For $t = 1, \dots, T$
 For $n = 1, \dots, N$
 For $i = 1, \dots, I$

$$L_FRF_{modt,n,i} = \begin{cases} \chi_{t,n}(e) & \text{if } i = \arg \max_{j, j=1, \dots, I} \{L_FRF_{t,n,j}\} \\ 0 & \text{otherwise} \end{cases}$$

In addition, in this method a weight for each tree obtained is introduced by testing each individual tree with the OOB

dataset. Let $\bar{p} = (p_1, p_2, \dots, p_T)$ be the vector with the weights assigned to each tree. Each p_t is obtained as $\frac{N_success_OOB_t}{size_OOB_t}$ where $N_success_OOB_t$ is the number of examples classified correctly from the OOB dataset used for testing the t -th tree and $size_OOB_t$ is the total number of examples in this dataset.

Strategy 1 → method MWLT1

The function $Faggre1_1$ is defined as:

$$Faggre1_1(t, i, L_FRF) = \begin{cases} 1 & \text{if } i = \arg \max_{j, j=1, \dots, I} \left\{ \sum_{n=1}^{N_t} L_FRF_{modt,n,j} \right\} \\ 0 & \text{otherwise} \end{cases}$$

Vector $\bar{p}$ is used in the definition of function $Faggre1_2$:

$$Faggre1_2(i, T_FRF) = \sum_{t=1}^T p_t \cdot T_FRF_{t,i}$$

Strategy 2 → method MWLT2

The vector of weights $\bar{p}$ is applied to Strategy 2.

$$Faggre2(i, L_FRF) = \sum_{t=1}^T p_t \sum_{n=1}^{N_t} L_FRF_{t,n,i}$$

REFERENCES

- [1] H. Ahn, H. Moon, J. Fazzari, N. Lim, J. Chen and R. Kodell, "Classification by ensembles from random partitions of high dimensional data," Computational Statistics and Data Analysis, vol. 51, pp. 6166–6179, 2007.
- [2] A. Fernández, M. J. del Jesus and F. Herrera, "Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets," Int. J. Approx. Reasoning, vol. 50, no. 3, pp. 561–577, 2009.
- [3] R. O. Duda, P. E. Hart and D. G. Stork, Pattern classification. John Wiley and Sons, Chichester, 2001.
- [4] D. J. C. Mackay, Information theory, inference and learning algorithms. Cambridge University Press, Cambridge, 2003.
- [5] J. R. Quinlan, C4.5: Programs for machine learning. Morgan Kaufmann Publishers, California, 1993.
- [6] I. H. Witten and E. Frank E, Data mining. Morgan Kaufmann Publishers, San Francisco, 2000.
- [7] P. P. Bonissone, "Approximate reasoning systems: handling uncertainty and imprecision in information systems," in Uncertainty Management in Information Systems: From Needs to Solutions, A. Motro and Ph. Smets, Eds. Kluwer Academic Publishers, 1997, pp. 369–395.
- [8] J. Casillas and L. Sánchez, "Knowledge extraction from data fuzzy for estimating consumer behavior models," IEEE conference on Fuzzy Systems, pp. 164–170, 2006.
- [9] M. C. Garrido, J. M. Cadenas and P. P. Bonissone, "A classification and regression technique to handle heterogeneous and imperfect information," Soft Computing, vol. 14, no. 11, pp. 1165–1185, 2010.
- [10] A. Otero, J. Otero, L. Sánchez and J. R. Villar, "Longest path estimation from inherently fuzzy data acquired with GPS using genetic algorithms," Int. Symposium on Evolving Fuzzy Systems, pp. 300–305, 2006.
- [11] P. P. Bonissone, J. M. Cadenas, M. C. Garrido and R. A. Díaz-Valladares, "A fuzzy random forest," Int. J. Approx. Reasoning, vol. 51, no. 7, pp. 729–747, 2010.
- [12] M. Umano, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu and J. Kinoshita, "Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems," IEEE Conference on Fuzzy Systems, vol. 3, pp. 2113–2118, 1994.
- [13] B. Sridevi and R. Nadarajan, "Fuzzy similarity measure for generalized fuzzy numbers," Int. J. Open Problems Compt. Math. vol. 2, no. 2, pp. 240–253, 2009.
- [14] A. M. Palacios, L. Sánchez and I. Couso I, "Extending a simple genetic cooperative-competitive learning fuzzy classifier to low quality datasets," Evolutionary Intelligence, vol. 2, pp. 73–84, 2009.
- [15] A. M. Palacios, L. Sánchez and I. Couso I, "Diagnosis of dyslexia with low quality data with genetic fuzzy systems," Int. J. Approx. Reasoning, vol. 51, pp. 993–1009, 2010.