Fuzzy Rule-Based Classification Systems for Multi-class Problems Using Binary Decomposition Strategies: On the Influence of n-dimensional Overlap Functions in the Fuzzy Reasoning Method

Mikel Elkano[a], Mikel Galar[a], Jose Sanz[a], Humberto Bustince[a,b]

[a] *Departamento de Automática y Computación, Universidad Pública de Navarra, 31006 Pamplona, Spain*
[b] *Institute of Smart Cities (ISC), Universidad Pública de Navarra, 31006 Pamplona, Spain*

Abstract

Multi-class classification problems appear in a broad variety of real-world problems, e.g., medicine, genomics, bioinformatics, or computer vision. In this context, decomposition strategies are useful to increase the classification performance of classifiers. For this reason, in a previous work we proposed to improve the performance of FARC-HD (Fuzzy Association Rule-based Classification model for High-Dimensional problems) fuzzy classifier using *One-vs-One* (OVO) and *One-vs-All* (OVA) decomposition strategies. As a result of an exhaustive experimental analysis, we concluded that even though the usage of decomposition strategies was worth to be considered, further improvements could be achieved by introducing n-dimensional overlap functions instead of the product t-norm in the Fuzzy Reasoning Method (FRM). In this way, we can improve confidences for the subsequent processing performed in both OVO and OVA.

In this paper, we want to conduct a broader study of the influence of the usage of n-dimensional overlap functions to model the conjunction in several Fuzzy Rule-Based Classification Systems (FRBCSs) in order to enhance their performance in multi-class classification problems applying decomposition techniques. To do so, we adapt the FRM of four well-known FRBCSs (CHI, SLAVE, FURIA, and FARC-HD itself). We will show that the benefits of the usage of n-dimensional overlap functions strongly depend on both the learning algorithm and the rule structure of each classifier, which explains why FARC-HD is the most suitable one for the usage of these functions.

Keywords: Fuzzy Rule-Based Classification Systems, Decomposition strategies, Overlap functions, Aggregations, One-vs-One, Multi-classification

Email addresses: mikel.elkano@unavarra.es (Mikel Elkano), mikel.galar@unavarra.es (Mikel Galar), joseantonio.sanz@unavarra.es (Jose Sanz), bustince@unavarra.es (Humberto Bustince)

1. Introduction

Fuzzy Rule-Based Classification Systems (FRBCSs) [34] are one of the most popular methods in pattern recognition and machine learning. These systems feature a good performance while providing interpretable models by using linguistic labels in the antecedents of their rules [34]. FRBCSs have been successfully applied to a wide variety of domains, including bioinformatics [26], medical problems [46], or financial applications [44], among others.

Within classification tasks, two types of problems can be identified depending on the number of classes considered: binary (two classes) and multi-class (more than two classes) problems. In general, the classifier learning is more difficult for multi-class problems. This is due to the increased complexity in the definition of decision boundaries, caused by the higher overlapping among the different classes of the problem. Even so, real-world problems need to consider multiple classes in many cases: for instance, arrhythmias classification [40], fingerprints recognition [23], or microarrays analysis [6]. In this context, the application of decomposition strategies [20, 39] is a straightforward manner for addressing multi-class problems, since they make any classifier capable of addressing these types of problems. Based on divide-and-conquer paradigm, the original multi-class problem is divided into easier-to-solve binary ones, which can be faced by independent binary classifiers called *base classifiers*.

Among decomposition strategies [39], *One-vs-One* (OVO) and *One-vs-All* (OVA) are the most common ones owing to their simplicity and accuracy. In the OVO scheme, the original problem is divided into as many binary sub-problems as possible pairs of classes, whereas in OVA as many sub-problems as classes in the original one are considered. When classifying a new instance, all base classifiers are queried and their outputs are combined to make the final decision (aggregation phase) [20]. These decomposition techniques usually obtain better results than addressing the problem directly, even when classifiers with inherent multi-class support are used [19, 20, 22, 43].

Previous works have shown the effectiveness of decomposition strategies when working with FRBCSs [15, 25, 30, 36]. Nevertheless, it should be borne in mind that, in these strategies, the final performance strongly depends on the outputs provided by each base classifier, since a new aggregation phase is introduced, which is not carried out when the problem is directly addressed. In our previous work [15], we showed that the outputs provided by FARC-HD (Fuzzy Association Rule-based Classification model for High-Dimensional problems) fuzzy classifier [2] were not suitable for decomposition schemes. This fact was due to the usage of the product to model the conjunction, since the aggregation of small values ended in outputs with low variation, quickly tending to zero. This effect was even more accentuated when the number of arguments (antecedents of fuzzy rules) increased, and as a consequence, those rules with more antecedents were penalized. However, these issues did not affect the baseline FARC-HD algorithm because output values were not used beyond the classification process. Otherwise, when using decomposition strategies, the previously mentioned facts became undesirable, since less knowledge was retained for the aggregation phase. Moreover, robust aggregations for OVO, such as weighted voting, obtained poor results with FARC-HD. On this account, the concept of n-dimensional overlap function was introduced in our previous work [15] with the aim of modeling the conjunction in the fuzzy rules of FARC-HD. In this manner, the values returned by base classifiers became more suitable for the aggregation phase, since they display a greater variation and they are independent of the number of arguments. This resulted in a significant increase in the final performance. Additionally, we proposed a new aggregation method for OVO (WinWV) with the aim of solving the problems of weighted voting caused by the unsuitable confidences provided by FARC-HD.

As a result of our previous work, the need for analyzing the behavior of n-dimensional overlap functions in different FRBCSs arises. More specifically, their behavior in the framework of multi-class problems using decomposition strategies must be analyzed. For this reason, in this paper we adapt the methodology presented in [15] to different FRBCSs. In order to obtain the broadest possible overview, we consider four different types of FRBCSs: Chi [12], SLAVE [25], FURIA [30], and FARC-HD [2] itself. We have selected these four classifiers as representative methods of FRBCSs since both their learning methods and their rule structure are clearly different. All of them have been adapted to use n-dimensional overlap functions in their Fuzzy Reasoning Method (FRM).

The main contributions of this work are the following:

- We analyze the performance of n-dimensional overlap functions in the four FRBCSs (CHI, SLAVE, FURIA, and FARC-HD) and we study whether the behavior shown in FARC-HD is extensible to other FRBCSs. In this manner, we aim to obtain a general overview of the behavior of these functions when they are applied to model the conjunction. Additionally, two decomposition strategies (OVO and OVA) are considered for each FRBCS.

- We study the impact of n-dimensional overlap functions on the rule bases generated in the four classifiers. As we will show, the usage of these functions does not only affect the performance of the model, but also its rule base. On this account, we analyze the average number of rules and antecedents per rule for each overlap function.

- We evaluate the performance of WinWV aggregation method (proposed to solve the problems of weighted voting with the confidences provided by FARC-HD) in the rest of FRBCSs. In order to do so, a comparison between WinWV aggregation strategy and the original weighted voting is performed considering the four FRBCSs.

In order to achieve well-founded conclusions, we carry out an empirical study considering twenty numerical datasets from the KEEL dataset repository [3] and we contrast the results obtained using non-parametric statistical tests, as suggested in the specialized literature [24]. In this study, we will analyze the influence of the usage of n-dimensional overlap functions when tackling directly the multi-class problem with FARC-HD, FURIA, CHI, and SLAVE baseline classifiers and when they are used as base classifiers for both OVO and OVA decomposition strategies. In all these cases, we have applied five different n-dimensional overlap functions and we have considered the usage of five aggregation strategies for OVO scheme.

The exhaustive analysis carried out shows that the benefit obtained is highly dependent on the learning process of each classifier, as well as on the structure of the rules generated after that process. The results obtained have allowed us to shed light on clarifying when it is appropriate to use n-dimensional overlap functions in the FRM of FRBCSs. That is, we explain why FARC-HD performs much better with these functions, whereas other FRBCSs present a rather different behavior.

The rest of this paper is organized as follows. Related works are reviewed in Section 2. In Section 3, we briefly describe the four FRBCSs considered in this work (FURIA, CHI, SLAVE, and FARC-HD) and we show their rule structure, learning and inference processes. Section 4 describes OVO and OVA decomposition strategies, along with the five aggregation strategies for OVO that we use in this paper. In Section 5, we recall the concept of n-dimensional overlap function and we describe the adaptation made to model the conjunction with these functions in each FRBCS considered. The experimental framework is presented in Section 6, whereas the analysis of the results obtained is given in Section 7. Finally, Section 8 concludes this paper.

2. Related Works

Fuzzy techniques are useful to achieve a trade-off between interpretability and accuracy in classification systems. In [1], authors developed a new approach to design fuzzy classifiers using k-means clustering and a memetic algorithm to find the optimal values of fuzzy rules and membership functions. Chen et al. [11] proposed a combination of a feature selection process applying modulator functions and a fuzzy rule extraction mechanism based on fuzzy clustering. In [38], authors presented a method to extract fuzzy rules from the sub-clusters produced by the output-interval clustering algorithm. Aliev et al. [4] extracted type-2 fuzzy rules applying fuzzy clustering and a Differential Evolution algorithm to optimize those rules. Finally, Sanz et al. [45] provided a framework to improve the performance of FRBCSs using interval-valued fuzzy sets.

Decomposition strategies can be considered as an ensemble method or a Multiple Classifier System (MCS), whose main objective is to enhance the classification performance using multiple classifiers. However, decomposition strategies focus on the usage of binary classifiers to address multi-class problems, whereas in ensembles and MCSs multi-class classifiers are usually considered in order to face such problems. This important difference has produced many different approaches for each type of method.

Ensemble techniques are traditionally based on creating diverse base classifiers that allow one to improve the performance as a result of the differences in their predictions, since they are complementary. Two of the most popular ensemble methods are Bagging [8] and Boosting [17], which have also been applied using fuzzy base classifiers [7, 35, 48]. In [7], authors proposed an extension of the classical Random Forests (a variant of bagging) making use of fuzzy decision trees. Ishibuchi and Nojima [35] combined the FRBCSs obtained in the Pareto front of a multi-objective optimization GA. In [48], authors developed a methodology to build MCSs using FURIA as base classifier, addressing all the stages from its construction (bagging-based) to the final combination process. These methods take advantage of the power of fuzzy systems to deal with soft decision boundaries, obtaining highly accurate models, but they may need thousands of rules [48]. These models are essentially focused on the final accuracy of the system, and therefore their interpretability is left aside. A clear example of this type of model is FURIA [30] (described in Section 3.4), which is one of the most extended base classifiers in this framework. FURIA generates adjusted hyper-rectangles for each rule instead of using the same linguistic labels for the entire rule base, and hence it cannot be considered as interpretable as a classical FRBCS [34]. For this reason, in this paper we will only consider decomposition-based ensembles, which may partially maintain the interpretability of the baseline models.

Decomposition strategies have become a commonly used approach to improve the performance of FRBCSs in multi-class classification problems [16, 30, 32, 47]. These strategies have been successfully applied using different base classifiers, such as Fuzzy Ripper [31], FH-GBML [36] or SLAVE [25] (described in Section 3.3). Moreover, Non-Dominance criterion (ND) [16] and Learning Valued Preference for Classification (LVPC) [30, 32] aggregation strategies (described in Section 4) have been specifically proposed for these fuzzy classifiers. In both of them, preference relations are considered to model the aggregation phase, where the best alternative should be predicted. In order to do so, Hüllermeier and Brinker [32] modeled the conflict and ignorance from the outputs of the Fuzzy Ripper algorithm [31]. From a different perspective, Fernandez et al. [16] proposed the usage of ND criterion in FH-GBML and SLAVE classifiers, obtaining good results. Finally, the Top-Down induction of Fuzzy Pattern Trees (PTTD) was presented in [47], where an OVA approach was applied.

In the framework of decomposition techniques, in [15] we proposed n-dimensional overlap functions to provide more suitable confidences when combining FARC-HD fuzzy classifier and decomposition strategies, which resulted in an enhancement of the final performance of FARC-HD. Based on this work, our aim is to extend this methodology to different FRBCSs and to study the behavior of these functions when they are applied in different FRMs.

3. Fuzzy Rule-Based Classification Systems

In this section we first introduce the preliminary concepts related to FRBCSs (Section 3.1). Next, a description of all the classifiers considered in this work is shown, along with their rule structure, learning algorithms and inference methods (Sections 3.2-3.5).

3.1. Preliminary concepts

In the literature, there are multiple techniques used to solve classification problems. Among them, FRBCSs are one of the most popular approaches, since they provide an interpretable model by means of the use of linguistic labels in their rules [34].

The two main components of FRBCSs are as follows.

1. *Knowledge base*: It is composed of both the rule base (RB) and the database, where the rules and the membership functions used to model the linguistic labels are stored, respectively.
2. *Fuzzy Reasoning Method (FRM)*: This is the mechanism used to classify examples using the information stored in the knowledge base.

In order to generate the knowledge base, a fuzzy rule learning algorithm is applied using a training set $\mathcal{D}_T$ composed of P labeled examples $x_p = (x_{p1}, \ldots, x_{pn}), p = \{1, \ldots, P\}$, where x_{pi} is the value of the i-th attribute ($i = \{1, 2, \ldots, n\}$) of the p-th training example. Each example belongs to a class $y_p \in \mathbb{C} = \{C_1, C_2, ..., C_m\}$, where m is the number of classes of the problem.

Since we consider multiple FRBCSs, in Table 1 we introduce the common notation to make them easier to understand.

Table 1: Notation defined for all FRBCSs considered in this paper.

Term	*Description*
n	number of variables
$\mathcal{D}_T$	training set
P	number of examples in the training set
x_p	p-th training example
$\mathbb{C}$	set of classes
m	number of classes
y_p	class of the p-th training example
R_j	j-th rule
n_j	number of antecedents of the j-th rule
C_j	class of the j-th rule
$\mathbb{L}_i$	set of linguistic labels for the i-th variable
l	number of linguistic labels in $\mathbb{L}_i$

3.2. CHI algorithm

CHI algorithm [12] generates the rule base establishing an association between variables (antecedents) and classes (consequents). The rule structure used by this algorithm is as follows.

$$\text{Rule } R_j : \text{ If } x_1 \text{ is } A_{j1} \text{ and } \dots \text{ and } x_n \text{ is } A_{jn} \text{ then Class} = C_j \text{ with } RW_j \tag{1}$$

where R_j is the label of the j-th rule, $x = (x_1, \dots, x_n)$ is a n-dimensional pattern vector that represents the example, $A_{ji} \in \mathbb{L}_i$ is a linguistic label modeled by a triangular membership function (being $\mathbb{L}_i = \{L_{i1}, \dots, L_{il}\}$ the set of linguistic labels for the i-th antecedent, where l is the number of linguistic labels in this set), C_j is the class label and RW_j is the rule weight computed using the most common specification, i.e., the fuzzy confidence value or certainty factor defined in [36]:

$$RW_j = CF_j = \frac{\displaystyle\sum_{x_p \in ClassC_j} \mu_{A_j}(x_p)}{\displaystyle\sum_{p=1}^{P} \mu_{A_j}(x_p)} \tag{2}$$

being $\mu_{A_j}(x_p)$ the matching degree of the example x_p with the antecedent part of the fuzzy rule R_j computed as follows:

$$\mu_{A_j}(x_p) = T\left(\mu_{A_{j1}}(x_{p1}), \dots, \mu_{A_{jn}}(x_{pn})\right) \tag{3}$$

where $\mu_{A_{ji}}(x_{pi})$ is the membership degree of the value x_{pi} to the fuzzy set A_{ji} of the rule R_j and T is a t-norm.

In order to construct the rule base, CHI applies the following learning process:

1. *Definition of the linguistic partitions.* Fuzzy partitions are constructed with the same triangular shape and equally distributed on the range of values.
2. *Generation of a fuzzy rule for each example.* A fuzzy rule is generated for each example x_p as follows.
   (a) The membership degree of each value x_{pi} to all the different fuzzy sets of the i-th variable is computed.
   (b) For each variable, the linguistic label with the greatest membership degree is selected.
   (c) A rule is generated for the example where the antecedent part is determined by the selected fuzzy region, that is, the intersection of the selected linguistic labels, and the consequent is the class label of the example (y_p). Notice that in this algorithm no feature selection is performed in the learning process, and hence all rules have exactly the same number of antecedents as variables in the problem (n).
   (d) The rule weight is computed using the certainty factor given in Eq. (2).

Note that after the learning process we can obtain duplicated rules with the same antecedent part and different consequent part. In that case, only the one with the highest rule weight is kept.

In order to classify a new example x_p, in this paper we consider the usage of the *additive combination* [14] FRM, which is composed of the following steps.

1. *Matching degree.* The strength of activation of the antecedent part for all rules in the rule base with the example x_p is computed (Eq. (3)).
2. *Association degree.* The association degree of the example x_p with each rule in the rule base is computed.
$$b_j(x_p) = \mu_{A_j}(x_p) \cdot RW_j \tag{4}$$
3. *Confidence degree.* The confidence degree for each class is computed. To obtain the confidence degree of a class, the association degrees of the rules of that class, i.e., those whose consequent is the class we are considering, are summed.
$$conf_c(x_p) = \sum_{R_j \in RB;\; C_j = c} b_j(x_p), \qquad c = 1, 2, \ldots, m \tag{5}$$
4. *Classification.* The class that obtains the highest confidence degree is predicted.
$$Class = arg \underset{c=1,\ldots,m}{max} (conf_c(x_p)) \tag{6}$$

3.3. SLAVE

SLAVE (Structural Learning Algorithm in a Vague Environment) [25] is an inductive learning algorithm that makes use of an iterative approach to learn fuzzy rules. In addition, it takes advantage of a Genetic Algorithm (GA) to reduce the number of rules, keeping only the most relevant ones for each class. The rule structure in SLAVE is as follows.

$$\text{Rule } R_j : \text{ If } x_1 \text{ is } \mathcal{A}_{j1} \text{ and } \ldots \text{ and } x_{n_j} \text{ is } \mathcal{A}_{jn_j} \text{ then Class} = C_j \text{ with } RW_j \tag{7}$$

where $\mathcal{A}_{ji} \subseteq \mathbb{L}_i$ is a subset of linguistic labels modeled by triangular membership functions and n_j is the number of antecedents of the rule. In this case, the rule weight is computed as:

$$RW_j = \frac{n^+(R_j)}{n(R_j)} \tag{8}$$

being $n^+(R_j)$ the number of positive examples for the rule R_j and $n(R_j)$ the number of covered examples by the rule R_j (the definition of these concepts is described in detail in [25]). A short example is shown below in order to clarify these types of rules.

Example 1. *A rule such as*

$$R_j : \textit{ If } x_1 \textit{ is } \{L_{11}, L_{13}, L_{14}\} \textit{ and } \ldots \textit{ and } x_{n_j} \textit{ is } \left\{L_{n_j2}, L_{n_j4}\right\} \textit{ then Class} = C_j \textit{ with } RW_j$$

is equivalent to

$$R_j : \textit{ If } (x_1 \textit{ is } L_{11} \textit{ or } x_1 \textit{ is } L_{13} \textit{ or } x_1 \textit{ is } L_{14}) \textit{ and } \ldots \textit{ and } \left(x_{n_j} \textit{ is } L_{n_j2} \textit{ or } x_{n_j} \textit{ is } L_{n_j4}\right)$$
$$\textit{then Class} = C_j \textit{ with } RW_j$$

As it can be observed, there are two main differences between CHI and SLAVE regarding the rule structure. On the one side, in SLAVE the number of antecedents may vary depending on the rule (an embedded feature selection process is carried out), whereas in CHI the number of antecedents in all rules is the same (all variables are used). On the other hand, in the case of SLAVE, a single antecedent can be composed of multiple linguistic labels, while in CHI each antecedent is a single linguistic label. In order to compute the disjunction (OR operator) of linguistic labels, the membership degrees of the input value to all of them are computed. Then, the maximum of these membership degrees is taken.

As in the case of CHI, the learning algorithm of SLAVE tries to obtain a rule base that represents the relationship between antecedents and the class, keeping only those antecedents that are necessary to properly represent the class for each rule. In order to do so, SLAVE applies an iterative method for each class in $\mathbb{C}$ that works as follows.

1. Given a training set $\mathcal{D}_T$ and a class C, the algorithm selects the best rule that represents the examples belonging to C. A rule is considered to be the best if it:

   - Covers the maximum number of examples of the class C.
   - Covers the minimum number of examples of the rest of classes.

   In order to find the best rule, SLAVE applies a Genetic Algorithm (GA) to simultaneously optimize both previous criteria.
2. The examples covered by the selected rule are removed from $\mathcal{D}_T$.
3. The process is repeated until no useful rules can be extracted for the class C. This situation happens when the optimization criteria cannot be fulfilled.
4. Once all rules for a class have been extracted, the same process is repeated with the rest of classes.

In order to classify a new example x_p, the inference works as follows:

1. *Adaptation degree.* The adaptation degree between the example and the antecedent part of each rule ($U_j(x_p, A_j)$) is computed. To do so, the measures of possibility of all A_{ji} are aggregated by a t-norm (in this case the product).

   $$U_j(x_p, A_j) = T\left(Poss(A_{j1}|x_{p1}), Poss(A_{j2}|x_{p2}), \ldots, Poss(A_{jn}|x_{pn_j})\right) \tag{9}$$

   The possibility measure of a given antecedent ($Poss(A_{ji}|x_{pi})$) is defined as the proportion of the maximum membership degree of the considered linguistic labels for that antecedent with respect to the maximum membership degree of all linguistic labels. The complete definition of this measure is presented in [25].
2. *Association degree.* The association degree of the example x_p with each rule in the rule base is computed.

   $$b_j(x_p) = U_j(x_p, A_j) \cdot RW_j \tag{10}$$

3. *Classification.* The class of the rule with the highest association degree ($b_j(x_p)$) is predicted. If there are two or more rules with the same association degree, SLAVE applies the following criteria:
   (a) The rule with the highest rule weight is the winner.
   (b) If the rule weights are the same, the rule that covered the least number of examples is the winner (in favor of specific rules).
   (c) In case of a tie, the first learned rule is the winner.

3.4. FURIA

FURIA (Fuzzy Unordered Rule Induction Algorithm) [30] modifies and extends RIPPER rule induction algorithm [13], learning fuzzy rules instead of conventional rules and unordered rule sets instead of rule lists. The rule structure in FURIA is as follows.

$$\text{Rule } R_j: \text{ If } x_1 \text{ is } A^I_{j1} \text{ and } \dots \text{ and } x_{n_j} \text{ is } A^I_{jn_j} \text{ then Class} = C_j \text{ with } RW_j \tag{11}$$

where A^I_{ji} is a trapezoidal membership function corresponding to the variable i defined as $A^I_{ji} = (\phi^{s,L}_{ji}, \phi^{c,L}_{ji}, \phi^{c,U}_{ji}, \phi^{s,U}_{ji})$, being $\phi^{c,L}_{ji}$ and $\phi^{c,U}_{ji}$ the lower and upper bounds of the core and $\phi^{s,L}_{ji}$ and $\phi^{s,U}_{ji}$ the lower and upper bounds of the support, respectively. With the aim of obtaining more flexible decision boundaries, FURIA applies a certainty factor to each rule (similar to the rule weight of SLAVE and CHI), which is computed as:

$$RW_j = \frac{2\dfrac{\left|\mathcal{D}_T^{(C_j)}\right|}{|\mathcal{D}_T|} + \displaystyle\sum_{x_p \in \mathcal{D}_T^{(C_j)}} \mu_{A^I_j}(x_p)}{2 + \displaystyle\sum_{x_p \in \mathcal{D}_T} \mu_{A^I_j}(x_p)} \tag{12}$$

where $\mathcal{D}_T$ represents the training set, $\mathcal{D}_T^{(C_j)}$ are the examples of the class of the rule (C_j), and $\mu_{A^I_j}(x_p)$ is the *coverage degree* (equivalent to the matching degree of Eq. (3)) of the rule R_j for the example x_p computed as:

$$\mu_{A^I_j}(x_p) = T\left(\mu_{A^I_{j1}}(x_{p1}), \mu_{A^I_{j2}}(x_{p2}), \dots, \mu_{A^I_{jn_j}}(x_{pn_j})\right) \tag{13}$$

being $\mu_{A^I_{ji}}(x_{pi})$ the membership degree of the i-th element and T a t-norm (in this case the product).

Looking at the rules used in FURIA, it can be observed that antecedents are not represented by triangular membership functions as in CHI and SLAVE. Instead, FURIA uses fuzzy sets with trapezoidal membership functions. We must stress that each membership function is specific to each antecedent, and thus it can be different for each fuzzy rule.

In order to generate the rule base, FURIA applies a learning algorithm composed of the following stages:

1. *Learn a rule set for each class using RIPPER algorithm.* This stage is divided into the building and optimization phases.

2. *Fuzzification of rules generated by RIPPER.* In this stage the structure of the rules is maintained, but the interval representing each antecedent is replaced by a trapezoidal membership function (Eq. (11)). To do so, the original interval of an antecedent is considered as the core ($\phi_{ji}^{c,L}$, $\phi_{ji}^{c,U}$) of the trapezoidal membership function, and then the optimal support bounds are adjusted. In order to solve this optimization problem, FURIA applies a greedy algorithm (in each rule) where a single antecedent i is fuzzified in each iteration, measuring the quality of that fuzzification in terms of *rule purity.* For this computation, only the relevant training data for rule j and antecedent i (D_T^{ji}) are considered:

   $$\mathcal{D}_T^{ji} = \left\{x \in \mathcal{D}_T \mid \mu_{A^I_{jk}}(x_k) > 0 \text{ for all } k = 1, \dots, n_j \text{ and } k \neq i\right\}$$

   Once the relevant data have been selected, this set is further divided into two subsets:

   - Positive instances (those belonging to the class of the rule), $\mathcal{D}_{T+}^{ji}$
   - Negative instances (rest of instances), $\mathcal{D}_{T-}^{ji}$

   Then, the rule purity is computed as follows:

   $$pur_{ji} = \frac{p_{ji}}{p_{ji} + n_{ji}} \tag{14}$$

   where

   $$p_{ji} = \sum_{x \in \mathcal{D}_{T+}^{ji}} \mu_{A^I_{j1}}(x_i)$$
   $$n_{ji} = \sum_{x \in \mathcal{D}_{T-}^{ji}} \mu_{A^I_{j1}}(x_i)$$

Note that after the fuzzification stage, each antecedent of each rule has its own trapezoidal membership function, and thus linguistic labels are not shared by all rules as in the rest of classifiers considered in this paper. Hence, FURIA makes use of fuzzy theory to improve the accuracy of the system, leaving its interpretability aside.

When classifying a new example x_p, FURIA applies the same FRM as CHI (Eq. (3)-(6)), but using trapezoidal membership functions ($\mu_{A^I_{ji}}(x_{pi})$) instead of triangular ones. In addition, if the example is not covered by any rule, a rule generalization process (*stretching*) is carried out replacing all rules by their minimal generalizations, which are obtained removing all the antecedents that are not satisfied by the query. In case of a tie, the class with highest frequency is predicted.

3.5. FARC-HD

FARC-HD (Fuzzy Association Rule-based Classification model for High-Dimensional problems) [2] is a fuzzy association rule-based classifier. Apriori algorithm is used to learn fuzzy rules before applying a subgroup discovery technique and an Evolutionary Algorithm is used to reduce the computational cost and improve the accuracy and interpretability of the model.

This method uses the following rule structure:

$$\text{Rule } R_j : \text{ If } x_1 \text{ is } A_{j1} \text{ and } \dots \text{ and } x_{n_j} \text{ is } A_{jn_j} \text{ then Class} = C_j \text{ with } RW_j \tag{15}$$

where the rule weight is computed applying the certainty factor (Eq (2)). As we can observe, the rule structure is the same as that of CHI (Eq. (1)). However, notice that in FARC-HD the number of antecedents may vary depending on the rules due to the way the latter are learned.

The learning algorithm of FARC-HD is composed of the three following stages:

1. *Fuzzy association rule extraction for classification*: In order to generate the rule base, a search tree is constructed for each class. To this end, frequent itemsets (sets of linguistic labels) are computed considering the support and confidence. Once the frequent itemsets are obtained, the fuzzy rules are extracted. The number of linguistic terms in the antecedents is limited by the maximum depth of the tree.
2. *Candidate rule pre-screening*: The most interesting fuzzy rules are pre-selected from the rule base obtained in the previous stage. To do so, a pattern weighting scheme is applied, where the weights of the examples are based on the coverage of the fuzzy rules.
3. *Genetic rule selection and lateral tuning*: An evolutionary algorithm is used both to tune the lateral position of the membership functions and to select the most accurate rules from the rule base generated in the previous steps.

In order to classify a new example, FARC-HD also applies the same FRM as CHI (Eq. (3)-(6)).

4. Decomposition strategies

Decomposition strategies [39] divide the original multi-class problem into simpler binary problems that are faced by independent binary classifiers, which are called base classifiers. These strategies are not only useful when working with classifiers that are only capable of discriminating between two classes, but also with those having an inherent multi-class support. Even in the latter case, the results are usually enhanced when decomposition strategies are applied [19, 20, 22, 43]. In this paper, we consider two of the most popular decomposition strategies in the literature: *One-Versus-One* (OVO) and *One-Versus-All* (OVA) [20] strategies.

4.1. One-Versus-One (OVO)

OVO strategy divides a m class problem into $m(m-1)/2$ binary sub-problems (all the possible combinations between pairs of classes). Each binary problem is faced by an independent base classifier which distinguishes a pair of classes $\{C_i, C_j\}$. When classifying a new example, all base classifiers are queried and their outputs are collected. For each classifier, a pair of confidence degrees $r_{ij}, r_{ji} \in [0, 1]$ in favor of classes C_i and C_j, respectively, are obtained. The outputs obtained from all base classifiers are stored in a *score-matrix* R:

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \tag{16}$$

Since each binary sub-problem is addressed by an independent base classifier, the score-matrix needs to be normalized in order to have all confidence degrees within the same range of values. This normalization is important when using classifiers that do not return confidences in [0,1], which could be interpreted as probabilities (which is the case of the FRBCSs tested in this paper). The normalization of the score-matrix ($\hat{R}$) is performed as follows.

$$\hat{r}_{ij} = \begin{cases} \dfrac{r_{ij}}{r_{ij} + r_{ji}} & \text{if } r_{ij} \neq 0 \text{ or } r_{ji} \neq 0 \\ 0.5 & \text{if } r_{ij} = r_{ji} = 0 \end{cases} \tag{17}$$

Finally, the outputs of base classifiers stored in the score-matrix are aggregated and the class is predicted. This aggregation phase is a key factor for the classification success [20]. Next, we briefly describe the five well-known OVO aggregation methods that we consider in this paper.

- *Voting strategy* (VOTE) [18]. Each base classifier gives a vote for its predicted class. The class having the largest number of votes is given as output:

  $$Class = arg \max_{i=1,\ldots,m} \sum_{1 \leq j \neq i \leq m} s_{ij} \tag{18}$$

  where s_{ij} is 1 if $\hat{r}_{ij} > \hat{r}_{ji}$ and 0 otherwise.

- *Weighted Voting* (WV) [33]. Each base classifier votes for both classes based on the confidence degree provided for each one. The class obtaining the highest value is given as output:

  $$Class = arg \max_{i=1,\ldots,m} \sum_{1 \leq j \neq i \leq m} \hat{r}_{ij} \tag{19}$$

- WinWV [15]. This aggregation method was proposed in our previous work [15] in order to solve the problems of WV with the confidences provided by FARC-HD. To do so, this method only considers the confidence of the predicted class, whereas that of the non-predicted class is not taken into account. Therefore, WinWV aggregation strategy works as follows:

  $$Class = arg \max_{i=1,\ldots,m} \sum_{1 \leq j \neq i \leq m} s_{ij} \tag{20}$$

  where s_{ij} is $\hat{r}_{ij}$ if $\hat{r}_{ij} > \hat{r}_{ji}$ and 0 otherwise.

- *Non-Dominance Criteria* (ND) [16]. The score-matrix is considered as a fuzzy preference relation. Then the non-dominance degree is computed, being the winning class the one with the highest value:

$$Class = arg \max_{i=1,\ldots,m} \left\{ 1 - \max_{j=1,\ldots,m} r'_{ji} \right\} \tag{21}$$

  where R' is the strict score-matrix (after normalization).

- *Learning valued preference for classification* (LVPC) [31, 32]. LVPC strategy considers the score-matrix as a fuzzy preference relation, as ND does. Based on fuzzy preference modeling, the original relation is decomposed into three new relations with different meanings: strict preference, conflict, and ignorance. Finally a decision rule based on a voting strategy is proposed to obtain the output class:

$$Class = arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} P_{ij} + \frac{1}{2} C_{ij} + \frac{N_i}{N_i + N_j} I_{ij} \tag{22}$$

  being N_i the number of training examples belonging to class i, C_{ij} the degree of conflict (the degree to which both classes are supported), I_{ij} the degree of ignorance (the degree to which none of the classes are supported), and P_{ij} and P_{ji} the strict preference for i and j, respectively. These variables are computed as follows:

$$C_{ij} = \min\{\hat{r}_{ij}, \hat{r}_{ji}\}, \qquad P_{ij} = \hat{r}_{ij} - C_{ij}, \qquad P_{ji} = \hat{r}_{ji} - C_{ij}, \qquad I_{ij} = 1 - \max\{\hat{r}_{ij}, \hat{r}_{ji}\}$$

It should be mentioned that, from the division of the multi-class problem in OVO, an inherent problematic issue arises: the non-competent classifiers [21]. This is due to the fact that each base classifier learns the model only using the examples belonging to the two classes that it discriminates, and thus the examples belonging to the rest of classes are ignored. Consequently, the remainder classes are unknown for this classifier and its outputs will be irrelevant to classify examples of those classes even though they are aggregated, since the non-competence cannot be established a priori. Even if this circumstance should be taken into account when applying OVO strategy, this problematic lies outside the scope of this paper and shall be considered in future works.

4.2. One-Versus-All (OVA)

OVA decomposition divides a m class problem into m binary sub-problems, which are faced by independent binary classifiers. Each base classifier distinguishes one of the classes from the remaining ones, learning the model using all examples of the training set. To this end, the examples of the class to be distinguished are considered as positives, whereas the rest are labeled as negatives. When classifying a new example, all base classifiers are queried and a confidence degree $\hat{r}_i \in [0, 1]$ in favor of the class C_i is returned by each classifier. The outputs of all base classifiers are stored in the *score-vector* R:

$$R = (r_1, \ldots, r_i, \ldots, r_m) \tag{23}$$

However, as in OVO, the range of the values returned by each classifier depends on each sub-problem. These differences among the ranges can lead us to misclassify an example, since the comparison among the confidences may not be fair. Therefore, the score-vector R needs to be normalized in such a way that all classifiers return values in the same range. To this aim, we normalize the score-vector with respect to the confidences obtained by each classifier for the negative class (stored in another score-vector $\overline{R}$). Once both vectors are obtained, the normalization of the score-vector ($\hat{R}$) is performed as follows.

$$\hat{r}_i = \frac{r_i}{r_i + \overline{r}_i} \tag{24}$$

Finally, in OVA the values of the score-vector are usually aggregated using the maximum, and thus the class with the highest confidence will be predicted. Another aggregation method for OVA is the so-called *dynamically ordered OVA* [29]. Nevertheless, in this work we only focus on the maximum because usually no statistical differences are found and the maximum is simpler.

5. Modeling the conjunction in FRBCSs with n-dimensional overlap functions: extending the FRMs

In our previous work [15], we showed that the confidences returned by FARC-HD are unsuitable for their subsequent processing in decomposition strategies. This was caused by the usage of the product in the FRM of FARC-HD. In order to solve this problem, we proposed to replace the product t-norm by n-dimensional overlap functions to model the conjunction in the FRM of FARC-HD. In this paper, we extend this methodology to four different FRBCSs by adapting their FRMs. In this manner, we aim to obtain a broader view of how n-dimensional overlap functions behave when they are used to model the conjunction in different FRBCSs.

In the rest of this section, we first recall the concept of n-dimensional overlap function introduced in our previous work [15] and we show the five different functions considered in this paper (Section 5.1). Next, we describe how these functions are included in the different FRBCSs in order to model the conjunction in their fuzzy rules (Section 5.2).

5.1. n-dimensional overlap functions

The original concept of overlap function [9] was introduced in image processing with the purpose of classifying those pixels whose belonging to the object or to the background was not clear. Examples of the application of these functions to image processing problems can be found in [37, 42]. Furthermore, these functions were also applied to model the indifference in preference relations [10]. Due to the fact that overlap functions allow one to recover many of the characteristics of t-norms without imposing the associativity property, their application range has turned out to be much broader. Taking advantage of these properties, an extension of overlap functions was proposed in our previous work [15] in order to adapt the inference process of FARC-HD to decomposition strategies by modeling the conjunction with these functions. With this aim, we extended the original concept of two dimensional overlap function to any finite dimension n (recovering the original definition when $n = 2$).

Let us recall the definition of the original two dimensional case:

Definition 1. [9] *A function $O : [0,1] \times [0,1] \to [0,1]$ is an overlap function if satisfies the following conditions :*

1. $O(x,y) = O(y,x)$ *for all* $x, y \in [0,1]$.
2. $O(x,y) = 0$ *if and only if* $x \cdot y = 0$.
3. $O(x,y) = 1$ *if and only if* $x \cdot y = 1$.
4. *O is increasing.*
5. *O is continuous.*

Based on the previous definition, the following extension was proposed:

Definition 2. [15] *A n-dimensional function $O : [0,1]^n \to [0,1]$ with $n \geq 2$ is a n-dimensional overlap function if the following properties hold:*

1. *O is symmetric.*
2. $O(x_1, \ldots, x_n) = 0$ *if and only if* $\prod_{i=1}^{n} x_i = 0$.
3. $O(x_1, \ldots, x_n) = 1$ *if and only if* $\prod_{i=1}^{n} x_i = 1$.
4. *O is increasing.*
5. *O is continuous.*

Furthermore, a construction method for n-dimensional overlap functions using rational expressions was presented:

Theorem 1. [15] *The mapping $O^n : [0,1]^n \to [0,1]$ is a n-dimensional overlap function if and only if there exist $f, g : [0,1]^n \to [0,1]$ with*

$$O^n(x_1, \ldots, x_n) = \frac{f(x_1, \ldots, x_n)}{f(x_1, \ldots, x_n) + g(x_1, \ldots, x_n)}$$

where

1. *f and g are symmetric.*
2. *f is non-decreasing and g is non-increasing.*
3. $f(x_1, \ldots, x_n) = 0$ *if and only if* $\prod_{i=1}^{n} x_i = 0$.

4. *$g(x_1, \ldots, x_n) = 0$ if and only if $\prod_{i=1}^{n} x_i = 1$.*
5. *f and g are continuous.*

In this paper we have considered five different n-dimensional overlap functions:

- **Product (PROD)**: The returned value is the product of the input values. The original behavior of all FRBCSs considered in this paper are recovered.

$$O(x_1, \ldots, x_n) = \prod_{i=1}^{n} x_i \tag{25}$$

- **Minimum (MIN)**: Returns the minimum of the input values. This is a t-norm as well, but unlike the product, the returned value does not decrease when the number of arguments increases. The minimum is commonly used in FRBCSs.

$$O(x_1 \ldots, x_n) = \min(x_1, \ldots, x_n) \tag{26}$$

- **Harmonic mean (HM)**: The returned value is the harmonic mean of the input values if all of them are different from zero and 0 otherwise.

$$O(x_1, x_2, \ldots, x_n) = \begin{cases} \dfrac{n}{\frac{1}{x_1} + \ldots + \frac{1}{x_n}} & \text{if } x_i \neq 0, \quad \text{for all } i = 1, \ldots, n \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

- **Geometric mean (GM)**: Returns the geometric mean of the input values.

$$O(x_1, x_2, \ldots, x_n) = \sqrt[n]{\prod_{i=1}^{n} x_i} \tag{28}$$

- **Sine (SIN)**: This overlap function returns higher values than means. It is interesting to study the behavior of these types of functions for modeling the conjunction.

$$O(x_1, \ldots, x_n) = \sin\left(\frac{\pi}{2}\left(\prod_{i=1}^{n} x_i\right)^{\alpha}\right) \tag{29}$$

  where $\alpha \leq \dfrac{1}{2n}$. In the experiments carried out in Section 7, we take $\alpha = \dfrac{1}{2n}$.

According to the values returned, we can establish an order among overlap functions. An overlap function is considered greater than other one if, for every possible input data, the values returned by the first function are higher than those returned by the second one. Among the considered overlap functions, the smallest one is the product t-norm, which returns values with a lower variation than the remaining functions when aggregating small values and whose output decreases as the number of arguments increases. Then, we have the minimum, a t-norm whose behavior is not affected by the number of arguments. Next, the harmonic and geometric means are considered (in this order) as representatives of means that return higher values than t-norms [5]. Finally, the largest function is the SIN, which returns higher values than means. The different behaviors among the considered overlap functions give us a general overview in the experiments carried out in Section 7.

In [15], we showed that those overlap functions satisfying the idempotency property provide better results, that is,

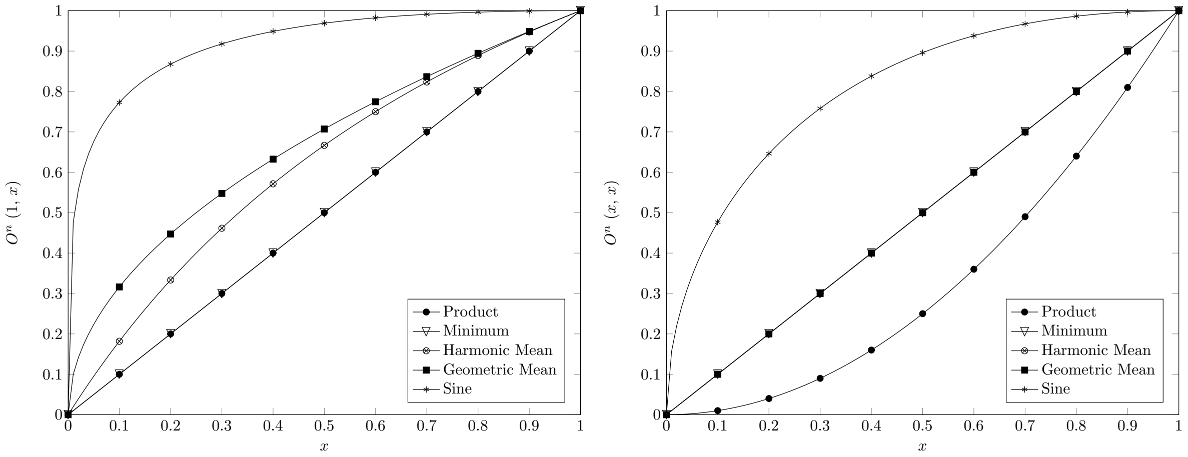
$$O(x, \dots, x) = x. \tag{30}$$

The reason is that the behavior of idempotent overlap functions is not affected by the number of antecedents. As we can observe, this property is satisfied by the minimum t-norm (Eq. (26)) and the harmonic (Eq. (27)) and geometric (Eq. (28)) means.

Fig. (1a) and (1b) show the previously mentioned differences in the behavior of the different overlap functions (we consider the two dimensional case, $n = 2$, to easily visualize their behavior). In Fig. (1a), we can observe the values returned by each overlap function when aggregating a value with 1, whereas Fig. (1b) depicts the returned values when aggregating a value with itself. Taking a look at Fig. (1a) and (1b), we can observe that the proposed n-dimensional overlap functions provide values with a higher variation than the product when aggregating small values. Nevertheless, both figures reveal a huge difference between the SIN and the rest of overlap functions, since the value returned by the SIN is greater than the input arguments when aggregating a value with itself (Fig. (1b)). We will experimentally show that this behavior may not be desirable in this framework, as it may produce a loss of discrimination power in the FRBCS. However, we have included this function aiming at obtaining a general overview of n-dimensional overlap functions and showing their behavior based on results.

5.2. Applying n-dimensional overlap functions in FRBCSs

One of the objectives of this work is to extend the usage of n-dimensional overlap functions introduced in [15] to other FRBCSs aiming at improving the performance when decomposition strategies are used. In this manner, we apply these functions to model the conjunction in fuzzy rules. As shown in the previous subsection, the aggregation of small values when using the product t-norm produces values with a low variation that tend quickly to 0. Moreover, when we consider FRBCSs where the number of antecedents can vary depending on the rule, this effect is even more accentuated in those rules with a higher number of antecedents. These factors have a different influence in baseline FRBCSs, as output values are not used beyond the classification. However, this is an undesirable circumstance when using decomposition strategies, since the knowledge acquired in the base classifiers is partially lost, producing a negative impact on the aggregation phase of these strategies.



(a) Considering $(1,\, x)$ as input

(b) Considering $(x,\, x)$ as input

Figure 1: Values returned by the different overlap functions.

In order to minimize the loss of knowledge and to obtain more suitable confidences when using decomposition strategies and FRBCSs, we propose to use n-dimensional overlap functions to model the conjunction in fuzzy rules. In this manner, the greater variation of the outputs of these functions and the fact that they are independent of the number of input arguments make confidences more suitable for the aggregation phase.

With the aim of studying the performance of n-dimensional overlap functions in multiple types of FRBCSs and obtaining the broadest possible overview, we have considered four different FRBCSs. The following is a detailed description of the application of overlap functions in each classifier.

5.2.1. Introducing n-dimensional overlap functions in the FRM of CHI

In this classifier, overlap functions replace the t-norm used in the matching and association degrees computation (Eq. (3) and (4), respectively), in the same way as it was done in [15] :

- *Matching degree*:

$$\mu_{A_j}(x_p) = O\left(\mu_{A_{j1}}(x_{p1}), \ldots, \mu_{A_{jn}}(x_{pn})\right) \tag{31}$$

- *Association degree*:

$$b_j(x_p) = O\left(\mu_{A_j}(x_p), RW_j\right) \tag{32}$$

As we described in Section 3.2, CHI algorithm does not perform any feature selection process and thus, all rules have exactly the same number of antecedents. Moreover, during the learning stage a rule is generated for each single instance in the training set, so the number of rules is the same when we use any overlap function. Therefore, we can observe that the usage of overlap functions does not have any major effect in the learning process, since neither the matching degree nor the association degree are considered for the generation of rules, except for rule weights computation (Eq. (2)).

5.2.2. Introducing n-dimensional overlap functions in the FRM of SLAVE

The computation of the association and adaptation degrees (Eq. (9) and (10)) is carried out by an overlap function, instead of the product:

- *Adaptation degree*:

$$U_j(x_p, A_j) = O\left(Poss(A_{j1}|x_{p1}), Poss(A_{j2}|x_{p2}), \ldots, Poss(A_{jn}|x_{pn_j})\right) \tag{33}$$

- *Association degree*:

$$b_j(x_p) = O\left(U_j(x_p, A_j), RW_j\right) \tag{34}$$

This algorithm (Section 3.3) carries out a feature selection process that is embedded into the learning stage, and therefore the number of antecedents may vary depending on the rule. In this case, the usage of overlap functions affects the learning process, and hence the number of antecedents and rules generated when using different overlap functions varies. The reason is that SLAVE makes use of the association and adaptation degrees during the feature and rule selection processes. Consequently, overlap functions are not only involved in the inference process but also have a direct influence on the rule base generated in the learning phase.

5.2.3. Introducing n-dimensional overlap functions in the FRM of FURIA

Overlap functions are applied in the same manner as in CHI (Eq. (31) and (32)), i.e., in the matching and association degrees (Eq. (3) and (4)). However, in this case the values to be aggregated are those returned by the different trapezoidal membership functions, instead of triangular ones.

FURIA (Section 3.4) learns all rules using RIPPER algorithm before fuzzifying them. This means that neither the number of antecedents nor the number of rules generated in the learning process depend on the overlap function used. Indeed, FURIA applies fuzzy sets theory after learning all rules in order to replace the interval of each antecedent by a trapezoidal membership function, but this fuzzification is performed only considering the purity of the rule (Eq. (14)), which is not computed using any t-norm. Thus, we can observe that overlap functions are not involved in the learning process of FURIA, except for the computation of rule weights (as in CHI).

5.2.4. Introducing n-dimensional overlap functions in the FRM of FARC-HD

The adaptation carried out in FARC-HD is the same as that performed in CHI (Eq. (31) and (32)). As in SLAVE, overlap functions are involved in all stages of the learning process of FARC-HD, since it makes use of both the matching and association degrees to extract the fuzzy rules and to perform the feature and rule selection processes. As we described in Section 3.5, this algorithm performs a lateral tuning of linguistic labels in order to improve the classification accuracy. Since the prediction is made using the matching and association degrees (Eq. (3)-(6)), the usage of overlap functions also affects the previously mentioned lateral tuning. This adjustment of membership functions helps FARC-HD to increase the benefits of overlap functions.

6. Experimental framework

This section is aimed at presenting the experimental framework setup used to carry out the experiments in Section 7, which is the same as that considered in [15]. First, we show the features of the datasets selected for the experimental study (Section 6.1). Next, the parameter setup considered for each method is described (Section 6.2). Finally, we introduce the performance measures and the statistical tests that are necessary to assess whether significant differences exist among the results obtained (Section 6.3).

6.1. Datasets

In order to test the performance of the different methods, we have considered twenty datasets selected from the KEEL dataset repository [3]. In Table 2, we find a summary of the features of all datasets, indicating for each one the number of examples (#Ex.), number of attributes (#Atts.), number of numerical (#Num.) and nominal (#Nom.) attributes, and the number of classes (#Class.).

All the experiments have been carried out using a *5-fold stratified cross-validation model*, i.e., we randomly split the dataset into five partitions of data, each one containing 20% of the examples, and we employed a combination of four of them (80%) to train the system and the remaining one to test it. Additionally, in each partition we consider three different seeds for the execution of the methods. Therefore, the result for each dataset is computed as the average of the five partitions using the three seeds in each one. In order to correct the dataset shift, that is, when the training data and the test data do not follow the same distribution, we will use a recently published partitioning procedure called *Distribution Optimally Balanced Cross Validation* [41], instead of the commonly used cross-validation.

6.2. Methods setup

Table 3 shows the configuration and parameters that we have considered for each FRBCS. The source code of all baseline classifiers was obtained from KEEL software [3]. The selected values are common for all problems, and they were selected according to the recommendation of the authors of each algorithm. Even though the tuning of parameters for each method on each particular problem could lead to better results, we preferred to maintain a baseline performance on each method as the basis for comparison, since we are not comparing algorithms among them.

Table 2: Summary of the features of the datasets used in the experimental study.

Id.	**Dataset**	**#Ex.**	**#Atts.**	**#Num.**	**#Nom.**	**#Class.**
aut	autos	159	25	15	10	6
bal	balance	625	4	4	0	3
cle	cleveland	297	13	13	0	5
con	contraceptive	1473	9	6	3	3
eco	ecoli	336	7	7	0	8
gla	glass	214	9	9	0	7
hay	hayes-roth	132	4	4	0	3
iri	iris	150	4	4	0	3
new	newthyroid	215	5	5	0	3
pag	pageblocks	548	10	10	0	5
pen	penbased	1100	16	16	0	10
sat	satimage	643	36	36	0	7
seg	segment	2310	19	19	0	7
shu	shuttle	2175	9	9	0	5
tae	tae	151	5	3	2	3
thy	thyroid	720	21	21	0	3
veh	vehicle	846	18	18	0	4
vow	vowel	990	13	13	0	11
win	wine	178	13	13	0	3
yea	yeast	1484	8	8	0	10

Table 3: Setup of the methods parameters.

Algorithm	*Parameters*
CHI	Num. of linguistic labels per variable: 3
	Rule weight: Certainty factor
SLAVE	Num. of linguistic labels per variable: 5
	Number of individuals: 100
	Mutation probability: 0.01
	Max. iterations without change: 500
FURIA	Num. of optimizations: 2
	Num. of folds: 3
FARC-HD	Num. of linguistic labels per variable: 5
	Minimum Support: 0.05
	Minimum Confidence: 0.8
	Maximum depth: 3
	Parameter k: 2
	Evaluations: 20000
	Number of individuals: 50
	α parameter: 0.02
	Bits per gen: 30
	Rule weight: Certainty factor

6.3. Performance measures and statistical tests

In this paper we have used the most common metric to test the performance of different methods, that is, the *accuracy rate*, which measures the percentage of correctly classified examples related to the total number of examples. However, we cannot extract well justified conclusions based only on the accuracy. For this reason, we apply some non-parametric tests [24] with the aim of providing statistical support to our results. Specifically, we use the Wilcoxon signed-ranks test [49] to perform pairwise comparisons, the Aligned Friedman test [27] to detect statistical differences among a group of methods, and the Holm *post-hoc* test [28] to find the algorithms that reject the null hypothesis of equivalence against the selected control method. A complete description of these tests can be found on the website: http://sci2s.ugr.es/sicidm/.

In addition to the previously mentioned performance measures, we also want to study the impact of overlap functions on the rule base. With this aim, we compute the average number of rules and antecedents per rule for each overlap function in both OVO and OVA models and in all baseline FRBCSs considered in this work. In the case of decomposition strategies, the average of all base classifiers is computed.

7. Experimental study

In this section, we study the results obtained by each method carrying out an analysis composed of four stages:

1. We test the performance of the different n-dimensional overlap functions when applying OVO and OVA models in all the FRBCSs considered in this paper (Section 7.1).
2. We study the impact of n-dimensional overlap functions on the rule base (Section 7.2).
3. We check whether the problems of WV with the confidences of FARC-HD are also present in the rest of FRBCSs considered in this paper comparing the original WV against WinWV (Section 7.3).
4. We discuss the results obtained in all previous points as a whole and we explain the reasons for the different behaviors in comparison with that obtained in FARC-HD (Section 7.4).

7.1. Analysis of the performance of n-dimensional overlap functions

Table 4 shows the average accuracy rate obtained in testing by the different FRBCSs (CHI, SLAVE, FURIA, and FARC-HD). As we can observe, we present the results obtained by each baseline FRBCS along with OVA scheme and with the five aggregation strategies of OVO model (ND, VOTE, LVPC, WV, WinWV). We show the performance of the five overlap functions (PROD, MIN, HM, GM, SIN) for each method, where the result of the best overlap function is highlighted in bold-face. These results are obtained by computing the average accuracy rate of each method in all datasets. The result of each dataset is computed as the average accuracy rate of the five partitions over the three different seeds.

Table 4: Average accuracy rate obtained in testing by each method.

	CHI					SLAVE					FURIA					FARC-HD				
	PROD	MIN	HM	GM	SIN	PROD	MIN	HM	GM	SIN	PROD	MIN	HM	GM	SIN	PROD	MIN	HM	GM	SIN
Baseline	**75.07**	71.20	67.41	66.92	65.74	**76.71**	74.37	74.31	74.54	73.08	**80.56**	80.55	80.46	80.49	80.12	**80.37**	80.17	80.11	79.89	79.98
OVA	**73.76**	67.39	64.20	63.36	61.93	**69.91**	69.13	68.97	68.47	64.85	80.39	80.38	80.40	80.36	**80.41**	79.92	80.27	**80.48**	80.13	79.97
OVOND	**77.10**	74.86	72.65	71.94	70.55	**77.41**	77.03	76.68	76.39	76.55	81.97	**81.98**	81.92	81.90	81.67	81.45	81.88	**82.18**	82.13	81.46
OVO^{VOTE}	**77.90**	75.55	73.72	73.05	71.98	**77.73**	77.34	77.17	76.72	77.06	82.37	**82.39**	82.37	82.34	82.12	81.52	82.03	**82.26**	82.25	81.71
OVO^{LVPC}	**77.93**	74.74	72.86	72.30	70.88	**71.72**	70.97	69.44	69.04	70.66	**82.52**	**82.52**	82.36	82.29	82.11	**79.77**	79.61	79.29	79.06	78.80
OVO^{WV}	**78.11**	75.31	73.42	72.75	71.17	**72.23**	71.63	69.73	69.33	71.79	**82.61**	82.58	82.45	82.38	82.20	80.19	80.16	**80.24**	80.05	78.96
OVO^{WinWV}	**78.19**	75.53	73.75	73.10	71.94	**76.07**	75.73	75.34	74.97	76.06	82.44	**82.45**	82.42	82.39	82.11	81.50	81.86	**81.93**	81.89	81.39

Additionally, in order to detect significant differences among the results of each overlap function in a given FRBCS, we carry out the Aligned Friedman test and the Holm post-hoc test, whose results are shown in Tables 5-8. The results of these tests are grouped in columns according to the method used to perform the comparison and in rows according to the overlap function considered. The first column corresponds to the baseline FRBCS execution applying each overlap function, whereas the second one shows the different overlap functions over OVA model. The rest of columns correspond to all OVO aggregation strategies considered in this paper (ND, VOTE, LVPC, WV and WinWV). The value of each cell corresponds to the rank obtained with the Aligned Friedman test when comparing the different overlap functions for each method (that is, an Aligned Friedman test is carried out for each group of methods in a column). The value shown in brackets indicates the adjusted p-value obtained by the Holm post-hoc test using as control method the one obtaining the lowest rank in the same column, which is shown in bold-face. The adjusted p-value is underlined when there are statistical differences ($\alpha = 0.1$ considering the ratio between datasets and algorithms).

Next, we explain the behavior of n-dimensional overlap functions in each baseline FRBCS, as well as when decomposition strategies are applied on them. We start describing the results obtained with FARC-HD, since n-dimensional overlap functions were first introduced in this FRBCS and we want to analyze the existing differences between the results obtained in this method with those in the remaining ones.

- **FARC-HD**
  - *Baseline*: as we can observe in Table 4, the five overlap functions considered in this paper obtain a similar performance. This is confirmed by the Aligned Friedman test shown in Table 5, since there are no statistical differences among them when executing the baseline FARC-HD algorithm. This means that FARC-HD is able to maintain the necessary classification accuracy when using overlap functions. The reason is that these functions are involved in all stages of the learning process (Section 5.2.4) and the generated rules are general enough to retain the discrimination

capability.

- *OVO and OVA models*: leaving the SIN aside, Tables 4 and 5 show that the greater the overlap function is, the better the results obtained are (although the GM is greater than the HM, both of them have a similar behavior). The problem with the SIN is that the value returned can be greater than all input values, which may not be a desirable behavior for an inference system because part of the discrimination capability is lost. Therefore, we observe that the best overlap functions in almost all cases are those returning the highest values preserving the idempotency property (HM and GM). Although the geometric and harmonic means return similar values, the latter one tends to obtain better results but without statistical differences. FARC-HD is able to take advantage of the confidences provided by these functions, since the classification accuracy is maintained when using overlap functions in the baseline model. Nevertheless, when LVPC and WV aggregations are used, the behavior of overlap functions changes due to the factors that will be described in Section 7.3. For this reason, a new aggregation strategy (WinWV) was presented in [15], which solved the problems of LVPC and WV with the confidences given by FARC-HD. This new aggregation method along with the problems of LVPC and WV when using FARC-HD are described in Section 7.3.

Table 5: Aligned Friedman and Holm tests to compare the different overlaps in FARC-HD, OVA and OVO.

	FARC-HD	OVA	OVO$^{\mathrm{ND}}$	OVO$^{\mathrm{VOTE}}$	OVO$^{\mathrm{LVPC}}$	OVO$^{\mathrm{WV}}$	OVO$^{\mathrm{WinWV}}$
PROD	**43.80**	57.90 (0.128)	55.23 (0.327)	56.53 (0.269)	**37.90**	**42.38**	54.40 (0.747)
MIN	48.63 (0.967)	51.72 (0.282)	49.03 (0.708)	49.77 (0.672)	41.22 (0.717)	42.95 (1.000)	46.42 (1.000)
HM	50.22 (0.967)	**38.23**	**40.52**	**40.95**	54.05 (0.157)	43.90 (1.000)	**43.83**
GM	56.25 (0.699)	48.95 (0.282)	45.65 (0.708)	43.65 (0.768)	56.67 (0.122)	49.13 (1.000)	47.95 (1.000)
SIN	53.60 (0.856)	55.70 (0.170)	62.08 (0.075)	61.60 (0.097)	62.65 (0.028)	74.15 (0.002)	59.90 (0.319)

- **SLAVE**
  - *Baseline*: looking at the accuracy rate (Table 4), we can observe that the product performs much better than the remaining overlap functions. This situation is confirmed in the statistical tests (Table 6), where there are significant differences in favor of the usage of the product when executing the baseline SLAVE. This classifier uses a different rule structure from that used in FARC-HD, which requires the greater discrimination capability provided by the product. There are some important differences between the rule structure of SLAVE and FARC-HD:
    1. The rules generated in SLAVE are more specific (with more antecedents) than in FARC-HD.
    2. As shown in Section 3.5, FARC-HD performs a lateral tuning in order to adjust the membership functions of fuzzy sets. As we stated, this adjustment is performed applying overlap

functions (Section 5.2.4), and hence the classifier accuracy optimization is carried out considering the overlap functions, increasing the benefits with respect to SLAVE.

3. The values of the antecedents in SLAVE are subsets of linguistic labels instead of single linguistic labels as in FARC-HD. For this reason, SLAVE needs to model the disjunction in fuzzy rules, which can cause a different effect when using overlap functions.

- *OVO model*: Table 4 shows that the accuracy obtained by all overlap functions when OVO model is considered is similar, although a decreasing trend can be observed as the overlap function increases. Looking at the statistical analysis in Table 6, we can observe that, even though accuracy rates are similar, there are statistical differences in favor of the product. However, it should be stressed that ranking differences between the product and the remaining overlap functions are reduced. This means that OVO takes advantage of the confidences returned by overlap functions, since the usage of these functions allows the performance of their respective base classifier to be raised in such a way that they obtain more similar results to that of the product. Therefore, the ratio of improvement of overlap functions in this model is greater than that of the product. The problem is that in this case the base classifiers do not provide enough classification accuracy (as it was shown in the baseline SLAVE) to obtain an improvement in OVO model when using overlap functions as in the case of FARC-HD.
- *OVA model*: as we can observe in Tables 4 and 6, contrary to the rest of the FRCBSs considered in this paper, the performance of this strategy is worse than that of the baseline SLAVE. This is due to the class imbalance problem that appears in this strategy and the inability of SLAVE to deal with this situation. Therefore, the behavior of overlap functions in this case is not representative in our framework.

Table 6: Aligned Friedman and Holm tests to compare the different overlaps in SLAVE, OVA and OVO.

	SLAVE	OVA	OVO$^{\mathrm{ND}}$	OVO$^{\mathrm{VOTE}}$	OVO$^{\mathrm{LVPC}}$	OVO$^{\mathrm{WV}}$	OVO$^{\mathrm{WinWV}}$
PROD	**21.47**	**37.02**	**33.63**	**32.40**	**30.95**	**30.55**	**40.42**
MIN	52.52 (0.001)	42.23 (1.000)	44.08 (0.255)	46.85 (0.115)	44.35 (0.144)	41.70 (0.224)	44.20 (0.681)
HM	56.63 (0.000)	41.05 (1.000)	53.90 (0.054)	51.10 (0.083)	61.27 (0.003)	66.10 (0.000)	53.48 (0.465)
GM	52.08 (0.001)	53.90 (0.198)	64.23 (0.003)	65.30 (0.001)	66.65 (0.000)	68.65 (0.000)	63.80 (0.043)
SIN	69.80 (0.000)	78.30 (0.000)	56.67 (0.036)	56.85 (0.023)	49.27 (0.092)	45.50 (0.206)	50.60 (0.535)

- **CHI**
  - *CHI*: taking a look at Table 4, we observe that the greater the overlap function is, the worse the results obtained are. This situation is confirmed by the Aligned Friedman and Holm post-hoc tests (Table 7), where we find significant differences in favor of the product. The reason for this behavior in comparison with FARC-HD is that the rules generated by CHI are much more specific than those of FARC-HD, since the number of antecedents is always equal to the number of features and they are learned considering all classes at the same time (whereas FARC-HD generates the rules class by class). Consequently, CHI algorithm needs more discrimination capability than FARC-HD due to the fact that the generated fuzzy rules are closer among themselves. Therefore, the usage of the product t-norm produces a greater discrimination power and leads to obtaining better results, whereas overlap functions highly affect the decision boundaries.
  - *OVO and OVA models*: Table 4 shows that, contrary to FARC-HD, the usage of overlap functions in the baseline CHI algorithm implies a loss of accuracy. Although in SLAVE this problem appears as well, the loss of accuracy in CHI is too great to obtain benefits from the confidences provided by overlap functions, as it is confirmed in the Aligned Friedman tests shown in Table 7. As a consequence, the behavior of these functions in OVO and OVA is the same as that observed in the baseline CHI.

Table 7: Aligned Friedman and Holm tests to compare the different overlaps in CHI, OVA and OVO.

	CHI	OVA	OVO^{ND}	OVO^{VOTE}	OVO^{LVPC}	OVO^{WV}	$\text{OVO}^{\text{WinWV}}$
PROD	**20.18**	**20.07**	**22.60**	**21.43**	**19.70**	**21.60**	**22.30**
MIN	28.00 (0.394)	35.92 (0.084)	33.02 (0.256)	33.65 (0.183)	35.45 (0.086)	32.63 (0.229)	34.50 (0.184)
HM	64.85 (0.000)	59.65 (0.000)	60.72 (0.000)	59.93 (0.000)	59.55 (0.000)	58.60 (0.000)	59.50 (0.000)
GM	67.38 (0.000)	65.57 (0.000)	63.55 (0.000)	65.45 (0.000)	63.68 (0.000)	64.03 (0.000)	64.58 (0.000)
SIN	72.10 (0.000)	71.27 (0.000)	72.60 (0.000)	72.05 (0.000)	74.13 (0.000)	75.65 (0.000)	71.63 (0.000)

- **FURIA**
  - *Baseline*: Tables 4 and 8 show that the behavior of all the overlap functions is similar in FURIA, and hence this algorithm is able to maintain the classification accuracy when using these functions (except for the SIN). In this case, overlap functions are not involved in any of the learning stages of FURIA. This is because rules are generated by RIPPER algorithm and t-norms are not used in the subsequent fuzzification process. Thus, the rules generated when using different overlap functions will be the same. Furthermore, FURIA uses highly adjusted trapezoidal membership functions (whose adjustment is not performed using t-norms) which provide high membership degrees, and hence the differences among the values returned by different overlap functions when aggregating large values are lower (Fig. 1a and 1b).
  - *OVO and OVA models*: as we can observe in Tables 4 and 8, both OVO and OVA models provide a similar performance when using different overlap functions. Even though FURIA maintains the classification accuracy when using overlap functions, the confidences provided by overlap functions are very similar due to the highly adjusted trapezoidal membership functions. The exception to this situation is when using the SIN overlap function due to the same reasons as those explained in the case of FARC-HD. In the same manner, LVPC and WV aggregation methods present different behaviors, which will be described in Section 7.3.

Table 8: Aligned Friedman and Holm tests to compare the different overlaps in FURIA, OVA and OVO.

	FURIA	OVA	OVO$^{\text{ND}}$	OVO$^{\text{VOTE}}$	OVO$^{\text{LVPC}}$	OVO$^{\text{WV}}$	OVO$^{\text{WinWV}}$
PROD	**40.17**	51.00 (1.000)	42.65 (0.871)	43.75 (1.000)	**38.25**	**34.65**	45.60 (1.000)
MIN	43.10 (0.750)	51.75 (1.000)	**40.90**	**42.77**	38.75 (0.956)	37.02 (0.796)	**41.60**
HM	50.90 (0.727)	50.00 (1.000)	48.05 (0.871)	44.50 (1.000)	52.53 (0.239)	50.78 (0.158)	45.22 (1.000)
GM	48.97 (0.727)	54.73 (1.000)	53.80 (0.479)	50.52 (1.000)	57.35 (0.112)	57.25 (0.041)	50.25 (1.000)
SIN	69.35 (0.006)	**45.03**	67.10 (0.018)	70.95 (0.008)	65.62 (0.011)	72.80 (0.000)	69.83 (0.008)

Summarizing, we can observe that the benefits of overlap functions are dependent on the learning process and the rule structure of each classifier. Therefore, the classifiers that are able to take advantage of overlap functions may be those which include these functions in their learning algorithms and have rules general enough to preserve the discrimination capability, maintaining the necessary classification accuracy. As we have shown, even though the confidences provided by overlap functions are more suitable for the aggregation performed in decomposition strategies, when the base classifiers do not provide enough classification accuracy these strategies do not obtain an improvement when using overlap functions.

Regarding decomposition strategies, Table 4 show their effectiveness when using FRBCSs, improving their performance in most of cases. However, the performance of OVA model can be affected by the increase in the imbalance ratio produced by the division performed in this strategy [20], as it occurs in SLAVE. We should keep in mind that in OVA each base classifier has to distinguish one of the classes from all others, and hence the proportion of instances of that class with respect to the rest of classes will be probably much smaller, particularly in datasets with a high number of classes (even if the original dataset is balanced).

7.2. Impact of n-dimensional overlap functions on the rule base

This subsection is aimed at showing the impact of n-dimensional overlap functions on the rule base. Table 9 presents the average number of rules and antecedents per rule for each baseline FRBCS and for OVA and OVO models (using as base classifier the same FRBCS). These averages are computed in the same manner as in Table 4, that is, by computing the average of each method in all datasets. The result of each dataset is computed as the average of the five partitions over the three different seeds.

Table 9: Average number of rules and antecedents.

		avg. rules					avg. antecedents				
		PROD	MIN	HM	GM	SIN	PROD	MIN	HM	GM	SIN
CHI	Baseline	170.43	170.43	170.43	170.43	170.43	12.40	12.40	12.40	12.40	12.40
	OVA	155.49	155.49	155.49	155.49	155.49	12.40	12.40	12.40	12.40	12.40
	OVO	86.94	86.94	86.94	86.94	86.94	12.40	12.40	12.40	12.40	12.40
SLAVE	Baseline	18.47	17.15	16.37	16.75	17.38	4.33	3.64	6.71	6.75	7.32
	OVA	3.78	3.61	3.45	3.40	3.56	2.36	2.11	3.45	3.38	3.85
	OVO	4.14	4.05	3.95	3.94	4.06	2.51	2.22	3.74	3.84	4.47
FURIA	Baseline	16.54	16.54	16.54	16.54	16.54	2.76	2.76	2.76	2.76	2.76
	OVA	7.95	7.95	7.95	7.95	7.95	2.05	2.05	2.05	2.05	2.05
	OVO	4.50	4.50	4.50	4.50	4.50	1.58	1.58	1.58	1.58	1.58
FARC-HD	Baseline	32.67	35.70	40.15	41.11	46.30	2.34	2.38	2.44	2.44	2.47
	OVA	13.03	14.26	16.09	16.64	18.35	1.76	1.79	1.84	1.84	1.86
	OVO	8.55	9.72	11.28	11.72	12.58	1.61	1.63	1.66	1.66	1.69

Next, we analyze the effect of n-dimensional overlap functions on the rule base of each FRBCS:

- **FARC-HD**: Table 9 shows that the usage of a greater overlap function implies a growing trend in the number of rules. A higher number of rules is needed in order to maintain or improve the discrimination capability, since the aggregation of the membership degrees returns larger values. On the other side, the number of antecedents is similar in all overlap functions, even though there is an upward trend when using greater overlap functions.

- **SLAVE**: As we can observe in Table 9 and contrary to FARC-HD, when we use greater overlap functions the number of rules decreases, whereas the number of antecedents increases. This behavior can be produced by two factors:

1. Due to the fact that SLAVE uses disjunctions in their rules, the usage of a greater overlap function may imply an increase in the number of linguistic labels in the antecedents, instead of implying an increase in the number of rules as in FARC-HD.
2. Since the rules generated in SLAVE are more specific than those generated in FARC-HD, the discrimination capability can be partially lost when using greater overlap functions, requiring the usage of more antecedents in the rules in order to maintain the necessary discrimination power.

- **CHI**: Since this algorithm does not perform any feature selection process, all rules will have exactly the same number of antecedents (equal to the number of features of the problem), and thus the usage of overlap functions does not alter the number of antecedents of the rules, as it can be observed in Table 9. Furthermore, CHI algorithm generates a new rule for each example, and consequently overlap functions do not have any effect in the number of rules generated. Another consequence of generating a new rule for each example is that the number of rules is notably greater than in the rest of methods. It should be noted, however, that the number of rules is usually considerably lower than the number of examples, since multiple rules are removed due to conflicts.

- **FURIA**: As in CHI, overlap functions are not involved in the learning process of FURIA, and hence the rules generated are the same for all overlap functions, as it can be observed in Table 9.

With respect to the comparison between baseline and decomposition strategies, Table 9 clearly shows that the rule base becomes simpler when decomposition strategies are applied. This is because these strategies divide the original problem into easier-to-solve binary sub-problems, needing a lower number of rules and antecedents to solve each sub-problem. In the same manner, the rule base of the classifiers in OVO will be simpler than in OVA, since OVO scheme considers only the examples of two classes while OVA takes into account all examples in the training set. However, when OVA is applied with SLAVE, we observe that there are less rules and antecedents than in OVO. This is caused by the inability of this algorithm to deal with the increase in the imbalance ratio produced by the division performed in OVA (as mentioned in Section 7.1).

7.3. WinWV

As we showed in [15] and Section 7.1, WV and LVPC are severely affected by the poor quality of the confidences of the non-predicted classes provided by FARC-HD, which is accentuated in LVPC due to the difficulty in modeling the conflict and ignorance terms. We focused on solving the problems of WV with the confidence of the non-predicted class due to the fact that if the conflict and ignorance terms were not considered in LVPC, the original WV would be recovered [15]. In order to solve this problem, WinWV was proposed (described in Section 4.1).

In this section, we check whether this situation is also present in the remaining FRBCSs. To do so, we carry out a number of pair-wise comparisons using the Wilcoxon signed-ranks test to confront the proposed aggregation method and the original WV, considering all FRBCSs used in this paper (CHI, SLAVE, FURIA, and FARC-HD) and the five different overlap functions. Table 10 shows the results of these comparisons, where R+ and R- indicate the ranks obtained by WinWV and WV, respectively.

As we can observe, WinWV aggregation strategy statistically outperforms the original WV method with all overlap functions in the case of SLAVE and FARC-HD. On the contrary, when considering CHI and FURIA algorithms there are no statistical differences between both aggregation methods in almost all cases. The reason is that in these two algorithms the confidence of the non-predicted class does not equally affect the aggregation in WV, since they are likely to be equal to 0. In the case of FURIA, this is due to the highly adjusted trapezoidal membership functions, whereas in CHI the reason is the high number of antecedents in its rules. The exception appears when overlap functions that are also t-norms are applied on FURIA, where WV performs better than WinWV. The confidences of the non-predicted class provided by FURIA are usually large when they are higher than 0 due to the highly adjusted trapezoidal membership functions. As a consequence, the usage of overlap functions makes the confidences of the non-predicted class to be increased much more quickly than those of the predicted one, and hence both confidences become more similar. Consequently, this algorithm loses discrimination capability and obtains worse results with WinWV.

7.4. Discussion

After analyzing the performance of n-dimensional overlap functions and their impact on the rule base, we have shown that the results obtained depend on each FRBCS. Additionally, the experimental study shows that the problems of WV with the confidences of FARC-HD are not present in all FRBCSs. For this reason, in this section we summarize and discuss all the previously mentioned points:

- *Performance of n-dimensional overlap functions*

  Even though the confidences provided by n-dimensional overlap functions are more suitable for the aggregation phase, decomposition strategies are not able to take advantage of these confidences if the base classifiers do not maintain the necessary classification accuracy when using these types of functions. This fact implies that the benefits obtained from the usage of n-dimensional overlap functions are strongly dependent on the learning algorithm and the rule structure of each FRBCS.

  In the case of FARC-HD, the baseline algorithm is able to maintain enough classification accuracy allowing decomposition strategies to take advantage of the confidences provided by overlap functions.

Table 10: Wilcoxon test to compare WinWV and WV.

FRCBS	WinWV vs. WV	R+	R-	p-value	Hypothesis
FARC-HD	PROD	203.00	7.00	0.000	Rejected for WinWV at 95%
	MIN	189.00	21.00	0.002	Rejected for WinWV at 95%
	HM	195.50	14.50	0.001	Rejected for WinWV at 95%
	GM	196.50	13.50	0.001	Rejected for WinWV at 95%
	SIN	204.50	5.50	0.000	Rejected for WinWV at 95%
SLAVE	PROD	197.50	12.50	0.001	Rejected for WinWV at 95%
	MIN	208.00	2.00	0.000	Rejected for WinWV at 95%
	HM	204.50	5.50	0.000	Rejected for WinWV at 95%
	GM	196.00	14.00	0.001	Rejected for WinWV at 95%
	SIN	203.50	6.50	0.000	Rejected for WinWV at 95%
CHI	PROD	107.00	103.00	0.981	Not rejected
	MIN	92.50	117.50	0.776	Not rejected
	HM	106.00	104.00	0.959	Not rejected
	GM	109.50	100.50	0.910	Not rejected
	SIN	164.00	46.00	0.044	Rejected for WinWV at 95%
FURIA	PROD	50.00	160.00	0.044	Rejected for WV at 95%
	MIN	67.00	143.00	0.179	Not rejected
	HM	97.00	113.00	0.836	Not rejected
	GM	123.50	86.50	0.532	Not rejected
	SIN	87.00	123.00	0.469	Not rejected

In SLAVE, despite the fact that baseline classifiers do not provide enough accuracy to obtain an improvement with respect to the product, the confidences obtained from overlap functions allow one to reduce these differences. On the other side, the baseline CHI algorithm dramatically loses discrimination capability when using overlap functions, and hence decomposition techniques are not able to exploit these functions. Finally, FURIA provides a similar performance with all overlap functions, since the confidences returned by this algorithm are too high to obtain an improvement from these functions.

- *Effect of n-dimensional overlap functions on the rule base*

  The usage of n-dimensional overlap functions not only affects the model performance, but also the rule bases. In the case of FURIA and CHI the size of the rule base is exactly the same with all overlap functions, since they are not involved in the rules generation process. However, in those algorithms where overlap functions are involved in the learning process (FARC-HD and SLAVE), the rule base is different depending on the overlap function used. In FARC-HD, the usage of a greater overlap function implies an increase in the number of rules, whereas the number of antecedents remains similar with all of them. Regarding SLAVE, greater overlap functions produce less rules but with greater number of antecedents.

  In all cases, the rule base of each classifier becomes simpler when decomposition strategies are used. Likewise, the rule bases in OVO are simpler than in OVA. The exception is the case of OVA with SLAVE, where the rule base is even simpler than in OVO due to the class imbalance produced by OVA scheme (described in Section 7.1).

- *WinWV*

  In the case of FARC-HD and SLAVE, WinWV performs much better than WV. However, the results obtained by WinWV and WV are similar when considering CHI and FURIA. The reason is that the confidences of the non-predicted class provided by these two algorithms are likely to be equal to 0.

All in all, we have shown that overlap functions improve the confidences of classifiers for the subsequent aggregation phase, but this improvement is only translated into a significant enhancement of the final performance if the baseline classifier is able to maintain the accuracy. In the rest of cases, the differences with respect to the product are reduced when decomposition strategies are used, which shows the benefits of overlap functions. However, a deeper analysis must be carried out in these cases in order to maintain the discrimination capability of the baseline classifiers while improving the confidences so that final accuracy could be boosted. At the same time, the rule base varies from one overlap function to another when these functions are involved in the learning process. Finally, in some FRBCSs the confidences obtained for the non-predicted class negatively affect the prediction in decomposition strategies, in which case WinWV is beneficial.

8. Conclusions

This work was motivated by the improvement found in FARC-HD when applying n-dimensional overlap functions and decomposition strategies. In this paper, we have carried out an exhaustive study that has allowed us to understand the influence of n-dimensional overlap functions in different FRBCSs.

In order to do so, we have studied whether the methodology presented in [15] improves the performance of four different FRBCSs (CHI, SLAVE, FURIA, and FARC-HD). As we have shown, the performance of overlap functions strongly depends on the learning process and rule structure of each classifier. Contrary to FARC-HD, CHI and SLAVE algorithms are not able to maintain the necessary discrimination capability when using overlap functions. Consequently, even though the confidences returned by overlap functions are more suitable for decomposition strategies, no improvement can be obtained from them. On the other side, FURIA is capable of preserving the classification accuracy when applying overlap functions, but the usage of highly adjusted trapezoidal membership functions implies that the membership degrees to be aggregated are likely to be 0 or close to 1. This produces small differences among the values returned by the different overlap functions, and hence they present similar behaviors in FURIA. In addition to the performance of overlap functions, we have analyzed their effect on the rule base of each FRBCS.

To sum up, after analyzing the behavior of overlap functions in four different FRBCSs, we can conclude that the performance of decomposition strategies will be significantly enhanced in those classifiers that involve the usage of overlap functions in their learning processes, maintaining the necessary discrimination capability and providing enough classification accuracy in the base classifiers.

There are several aspects that remain to be addressed in future works. Among them, the issue of non-competent classifiers [21] must be considered when working with OVO scheme. Furthermore, a more in depth study of the effect of decomposition strategies on the interpretability of FRBCSs should be carried out. Finally, the comparison and combination between decomposition-based techniques and preprocessing-based fuzzy ensembles, such as bagging [48], could also be studied. In this latter case, we would make use of fuzzy techniques with the unique aim of enhancing the classification performance, which is a completely different perspective than the one considered in this work.

Acknowledgment

This work has been supported by the Spanish Ministry of Science and Technology under the project TIN-2013-40765-P.

[1] Acilar, A., Arslan, A., 2014. A novel approach for designing adaptive fuzzy classifiers based on the combination of an artificial immune network and a memetic algorithm. Information Sciences 264, 158–181.

[2] Alcalá-Fdez, J., Alcalá, R., Herrera, F., 2011. A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. IEEE Transactions on Fuzzy Systems 19 (5), 857–872.

[3] Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F., 2011. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic and Soft Computing 17:2-3, 255–287.

[4] Aliev, R., Pedrycz, W., Guirimov, B., Aliev, R., Ilhan, U., Babagil, M., Mammadli, S., 2011. Type-2 fuzzy neural networks with fuzzy clustering and differential evolution optimization. Information Sciences 181 (9), 1591–1608.

[5] Beliakov, G., Pradera, A., Calvo, T., 2007. Aggregation Functions: A Guide for Practitioners. Vol. 221 of Studies in Fuzziness and Soft Computing. Springer Berlin Heidelberg.

[6] Bolón-Canedo, V., no, N. S.-M., Alonso-Betanzos, A., 2012. An ensemble of filters and classifiers for microarray data classification. Pattern Recognition 45 (1), 531–539.

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

[8] Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123–140.

[9] Bustince, H., Fernandez, J., Mesiar, R., Montero, J., Orduna, R., 2010. Overlap functions. Nonlinear Analysis: Theory, Methods & Applications 72 (3-4), 1488–1499.

[10] Bustince, H., Pagola, M., Mesiar, R., Hullermeier, E., Herrera, F., 2012. Grouping, overlap, and generalized bientropic functions for fuzzy modeling of pairwise comparisons. IEEE Transactions on Fuzzy Systems 20 (3), 405–415.

[11] Chen, Y., Pal, N. R., Chung, I., 2012. An integrated mechanism for feature selection and fuzzy rule extraction for classification. IEEE Transaction on Fuzzy Systems 20 (4), 683–698.

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

[13] Cohen, W. W., 1995. Fast effective rule induction. presented at the 12th Int. Conf. Mach. Learn., Lake Tahoe, CA, USA.

[14] Cordón, O., del Jesus, M. J., Herrera, F., 1999. A proposal on reasoning methods in fuzzy rule-based classification systems. International Journal of Approximate Reasoning 20 (1), 21–45.

[15] Elkano, M., Galar, M., Sanz, J., Fernández, A., Barrenechea, E., Herrera, F., Bustince, H., 2014. Enhancing multi-class classification in FARC-HD fuzzy classifier: On the synergy between n-dimensional overlap functions and decomposition strategies. IEEE Transactions on Fuzzy Systems.

[16] Fernández, A., Calderón, M., Barrenechea, E., Bustince, H., Herrera, F., 2010. Solving mult-class problems with linguistic fuzzy rule based classification systems based on pairwise learning and preference relations. Fuzzy Sets and Systems 161 (23), 3064–3080.

[17] Freund, Y., Schapire, R. E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55 (1), 119–139.

[18] Friedman, J., 1996. Another approach to polychotomous classification. Tech. rep., Department of Statistics, Stanford University.

[19] Fürnkranz, J., 2003. Round robin ensembles. Intelligent Data Analysis 7 (5), 385–403.

[20] Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F., 2011. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. Pattern Recognition 44 (8), 1761 – 1776.

[21] Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F., 2013. Dynamic classifier selection for one-vs-one strategy: Avoiding non-competent classifiers. Pattern Recognition 46 (12), 3412–3424.

[22] Galar, M., Fernández, A., Barrenechea, E., Herrera, F., 2014. Empowering difficult classes with a similarity-based aggregation in multi-class classification problems. Information Sciences 264, 135–157.

[23] Galar, M., Sanz, J., Pagola, M., Bustince, H., Herrera, F., 2014. A preliminary study on fingerprint classification using fuzzy rule-based classification systems. In: 2014 IEEE World Congress on Computational Intelligence (IEEE WCCI 2014) – 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2014).

[24] García, S., Fernández, A., Luengo, J., Herrera, F., 2010. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180 (10), 2044–2064.

[25] González, A., Perez, R., 1999. SLAVE: a genetic learning system based on an iterative approach. IEEE Transactions on Fuzzy Systems 7 (2), 176–191.

[26] Ho, S., Hsieh, C., Chen, H., Huang, H., 2006. Interpretable gene expression classifier with an accurate and compact fuzzy rule base for microarray data analysis. Biosystems 85 (3), 165–176.

[27] Hodges, J. L., Lehmann, E. L., 1962. Ranks methods for combination of independent experiments in analysis of variance. Ann. Math. Statist. 33, 482–497.

[28] Holm, S., 1979. A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6, 65–70.

[29] Hong, J., Min, J., Cho, U., Cho, S., 2008. Fingerprint classification using one-vs-all support vector machines dynamically ordered with Naïve Bayes classifiers. Pattern Recognition 41 (2), 662–671.

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

[31] Huhn, J. C., Hullermeier, E., 2009. FR3: A fuzzy rule learner for inducing reliable classifiers. IEEE Transactions on Fuzzy Systems 17 (1), 138–149.

[32] Hüllermeier, E., Brinker, K., 2008. Learning valued preference structures for solving classification problems. Fuzzy Sets and Systems 159 (18), 2337–2352.

[33] Hüllermeier, E., Vanderlooy, S., 2010. Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting. Pattern Recognition 43 (1), 128–142.

[34] Ishibuchi, H., Nakashima, T., Nii, M., 2004. Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining. Springer-Verlag.

[35] Ishibuchi, H., Nojima, Y., 2006. Fuzzy ensemble design through multi-objective fuzzy rule selection. In: Jin, Y. (Ed.), Multi-Objective Machine Learning. Vol. 16 of Studies in Computational Intelligence. Springer Berlin Heidelberg, pp. 507–530.

[36] Ishibuchi, H., Yamamoto, T., Nakashima, T., 2005. Hybridization of fuzzy GBML approaches for pattern classification problems. IEEE Transactions on System, Man and Cybernetics B 35 (2), 359–365.

[37] Jurio, A., Bustince, H., Pagola, M., Pradera, A., Yager, R. R., 2013. Some properties of overlap and grouping functions and their application to image thresholding. Fuzzy Sets and Systems 229, 69–90.

[38] Ling, W., Lu, W., 2014. Fuzzy rules extraction based on output-interval clustering and support vector regression for forecasting. Journal of Intelligent & Fuzzy Systems 27, 2563–2571.

[39] Lorena, A., Carvalho, A., Gama, J., 2008. A review on the combination of binary classifiers in multiclass problems. Artificial Intelligence Review 30 (1-4), 19–37.

[40] Melin, P., Amezcua, J., Valdez, F., Castillo, O., 2014. A new neural network model based on the lvq algorithm for multi-class classification of arrhythmias. Information Sciences 279, 483–497.

[41] Moreno-Torres, J., Saez, J., Herrera, F., 2012. Study on the impact of partition-induced dataset shift on k-fold cross-validation. IEEE Transactions on Neural Networks and Learning Systems 23 (8), 1304–1312.

[42] Paternain, D., Pagola, M., Fernandez, J., Mesiar, R., Beliakov, G., Bustince, H., 2011. Brain MRI thresholding using

incomparability and overlap functions. In: Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on. pp. 808–812.

[43] Sáez, J. A., Galar, M., Luengo, J., Herrera, F., 2014. Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition. Knowledge and Information Systems 38 (1), 179–206.

[44] Sanz, J., Bernardo, D., Herrera, F., Bustince, H., Hagras, H., 2014. A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data. IEEE Transactions on Fuzzy Systems.

[45] Sanz, J., Fernández, A., Bustince, H., Herrera, F., 2013. IVTURS: A linguistic fuzzy rule-based classification system based on a new interval-valued fuzzy reasoning method with tuning and rule selection. IEEE Transactions on Fuzzy Systems 21 (3), 399–411.

[46] Sanz, J., Galar, M., Jurio, A., Brugos, A., Pagola, M., Bustince, H., 2013. Medical diagnosis of cardiovascular diseases using an interval-valued fuzzy rule-based classification system. Applied Soft Computing JournalArticle in Press.

[47] Senge, R., Hüllermeier, E., 2011. Top-down induction of fuzzy pattern trees. IEEE Transactions on Fuzzy Systems 19, 241 – 252.

[48] Trawinski, K., Cordon, O., Sanchez, L., Quirin, A., 2013. A genetic fuzzy linguistic combination method for fuzzy rule-based multiclassifiers. IEEE Transactions on Fuzzy Systems 21 (5), 950–965.

[49] Wilcoxon, F., 1945. Individual comparisons by ranking methods. Biometrics 1 (6), 80–83.