



On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets

Alberto Fernández^{a,*}, María José del Jesus^b, Francisco Herrera^a

^a Department of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, Spain

^b Department of Computer Science, University of Jaén, Spain

ARTICLE INFO

Article history:

Received 2 November 2008

Received in revised form 23 November 2009

Accepted 21 December 2009

Keywords:

Fuzzy rule based classification systems

Linguistic 2-tuples representation

Tuning

Genetic algorithms

Genetic Fuzzy Systems

Imbalanced data-sets

ABSTRACT

When performing a classification task, we may find some data-sets with a different class distribution among their patterns. This problem is known as classification with imbalanced data-sets and it appears in many real application areas. For this reason, it has recently become a relevant topic in the area of Machine Learning.

The aim of this work is to improve the behaviour of fuzzy rule based classification systems (FRBCSs) in the framework of imbalanced data-sets by means of a tuning step. Specifically, we adapt the 2-tuples based genetic tuning approach to classification problems showing the good synergy between this method and some FRBCSs.

Our empirical results show that the 2-tuples based genetic tuning increases the performance of FRBCSs in all types of imbalanced data. Furthermore, when the initial Rule Base, built by a fuzzy rule learning methodology, obtains a good behaviour in terms of accuracy, we achieve a higher improvement in performance for the whole model when applying the genetic 2-tuples post-processing step. This enhancement is also obtained in the case of cooperation with a preprocessing stage, proving the necessity of rebalancing the training set before the learning phase when dealing with imbalanced data.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

There are many tools in the context of Machine Learning for solving a classification problem. One of them, known as fuzzy rule based classification systems (FRBCSs) [43], has the advantage of being easily interpretable by the end user or the expert. The disadvantage of these systems is their lack of accuracy when dealing with some complex systems, i.e. high dimensional problems, when the classes are overlapped or in the presence of noise, due to the inflexibility of the concept of linguistic variables, which imposes hard restrictions on the fuzzy rule structure [9].

In the specialized literature we can find different proposals to increase the accuracy of linguistic fuzzy systems, both applied to modeling and classification problems [1,12,21]. These approaches try to induce better cooperation among the rules by acting on one or two different model components: the fuzzy partition parameters stored in the Data Base (DB) and the Rule Base (RB).

To ease the genetic optimization of the DB membership functions (MFs), a new linguistic rule representation model was proposed in [2]. It is based on the linguistic 2-tuples representation [40] that allows the lateral displacement of a label considering a unique parameter. This way of working involves a reduction in the search space that eases the derivation of

* Corresponding author. Tel.: +34 958 240598; fax: +34 958 243317.

E-mail addresses: alberto@decsai.ugr.es (A. Fernández), mjjesus@ujaen.es (M.J. del Jesus), herrera@decsai.ugr.es (F. Herrera).

optimal models. In [2,3] the authors determined the high potential of this approach in regression problems, and our intention is to apply this genetic tuning to classification with imbalanced problems.

The problem of imbalanced data-sets [14], occurs when one class, usually the one that contains the concept to be learnt (the positive class), is underrepresented in the data-set. Addressing the class imbalance problem is a current challenge of the Data Mining community [72], and we must emphasize the significance of this situation since such types of data appears in most of the real domains of classification, i.e. risk management [42], medical diagnosis [54] and face recognition [52] among others.

Most learning algorithms obtain a high predictive accuracy over the majority class, but predict poorly over the minority class [67]. Furthermore, the examples of the minority class can be treated as noise and they might be completely ignored by the classifier. There are studies that show that most classification methods lose their classification ability when dealing with imbalanced data [47,57].

The aim of this study is to improve the results obtained by FRBCSs in imbalanced data-sets by means of the application of the 2-tuples based genetic tuning. We want to enhance the performance of our fuzzy model to make it competitive with C4.5 [59], a decision tree algorithm that presents a good behaviour in imbalanced data-sets [55,61,62], and with Ripper [17], a traditional and accurate rule based classifier algorithm. We will also show that we can obtain a fuzzy classification model with a lower complexity than the standard interval rule learning algorithms, together with an intrinsic higher interpretability because of the use of fuzzy labels, as we have stated at the beginning of this section.

In this paper we use two learning methods in order to generate the RB for the FRBCS. The first one is the method proposed in [16], that we have named the Chi et al.'s rule generation. The second approach is defined by Ishibuchi and Yamamoto in [45] and it consists of a Fuzzy Hybrid Genetic Based Machine Learning (FH-GBML) algorithm.

In our first study on the topic [33], we analysed the behaviour of FRBCSs looking for the best configuration of the fuzzy components and the synergy with preprocessing techniques to deal with the problem of imbalanced data-sets. According to the decisions taken in that work, in this paper we will use triangular membership functions for the fuzzy partitions and rule weights in the consequent of the rules. We will study the use of the 2-tuples tuning directly over the original data-sets using the appropriate measure of performance to guide the search, but we will also apply a re-sampling procedure as a solution at the data level to deal with the imbalance problem, specifically using the “Synthetic Minority Over-sampling Technique” (SMOTE) [13] to prepare the training data for the learning process.

The rest of this paper is organized as follows: In Section 2, we present the imbalanced data-set problem, describing the preprocessing technique used in our work, the SMOTE algorithm, and discussing the evaluation metrics. In Section 3, we describe the fuzzy rule learning methodologies used in this study. Next, Section 4 shows the significance of the tuning of the fuzzy systems and introduces the 2-tuples tuning approach and the evolutionary algorithm that tunes the FRBCS. In Section 5, we include our experimental analysis in imbalanced data-sets with different degrees of imbalance, where we compare the FRBCSs with 2-tuples based genetic tuning with Ripper and C4.5, in order to validate our results. In Section 6, some concluding remarks and suggestions for further work are made. Finally, we include an appendix with the detailed results for the experiments performed in the experimental study.

2. Imbalanced data-sets in classification

In this section, we will first introduce the problem of imbalanced data-sets. Then, we will describe the preprocessing technique we have applied in order to deal with the imbalanced data-sets: the SMOTE algorithm. Finally, we will present the evaluation metrics for this type of classification problem.

2.1. The problem of imbalanced data-sets

In some classification problems, the number of instances of every class is different. Specifically when facing a data-set with only two classes, the imbalance problem occurs when one class is represented by a large number of examples, while the other is represented by only a few [14].

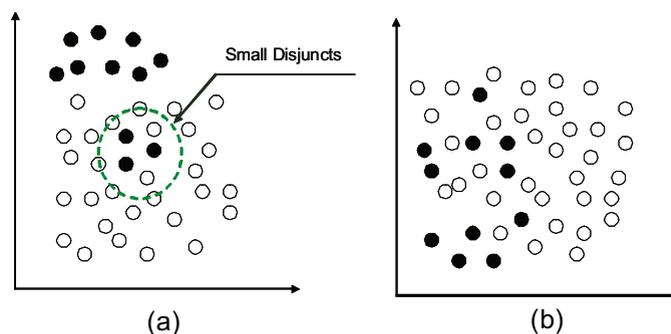


Fig. 1. Example of the imbalance between classes: (a) small disjuncts and (b) overlapping between classes.

The problem of imbalanced data-sets is extremely significant [72] because it is implicit in most real world applications, such as satellite image classification [64], risk management [42], protein data [58] and particularly in medical applications [49,54,56]. It is important to point out that the minority class usually represents the concept of interest, for example patients with illnesses in a medical diagnosis problem; while the other class represents the counterpart of that concept (healthy patients).

Standard classifier algorithms have a bias towards the majority class, since the rules that predict the higher number of examples are positively weighted during the learning process in favour of the accuracy metric. Consequently, the instances that belong to the minority class are misclassified more often than those belonging to the majority class. Another important issue related to this type of problem is the presence of small disjuncts in the data-set [66] and the difficulty most learning algorithms have in detecting those regions. Furthermore, the main handicap in imbalanced data-sets is the overlapping between the examples of the positive and the negative class [36]. These facts are depicted in Fig. 1a and b, respectively.

In the specialized literature, researchers usually manage all imbalanced data-sets as a whole [8,10,15]. Nevertheless, in this paper we will organize the different data-sets according to their degree of imbalance using the imbalance ratio (IR) [55], which is defined as the ratio of the number of instances of the majority class and the minority class.

A large number of approaches have been previously proposed to deal with the class-imbalance problem. These approaches can be categorized in two groups: the internal approaches that create new algorithms or modify existing ones to take the class-imbalance problem into consideration [8,28,70,71] and external approaches that preprocess the data in order to diminish the effect of their class imbalance [10,30]. Furthermore, cost-sensitive learning solutions incorporating both the data and algorithmic level approaches assume higher misclassification costs with samples in the minority class and seek to minimize the high cost errors [25,63,73].

The great advantage of the external approaches is that they are more versatile, since their use is independent of the classifier selected. Furthermore, we may preprocess all data-sets beforehand in order to use them to train different classifiers. In this manner, the computation time needed to prepare the data is only required once.

In our previous work on this topic [33], we analysed the cooperation of some preprocessing methods with FRBCSs, showing a good behaviour for the over-sampling methods, especially in the case of the SMOTE methodology [13]. In accordance with these results, we will use the SMOTE algorithm in this paper in order to deal with the problem of imbalanced data-sets, which is detailed in the next subsection.

2.2. Preprocessing imbalanced data-sets. The SMOTE algorithm

As mentioned before, applying a preprocessing step in order to balance the class distribution is a positive solution to the imbalanced data-set problem [10]. Specifically, in this work we have chosen an over-sampling method which is a reference in this area: the SMOTE algorithm [13].

In this approach, the positive class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbours. Depending upon the amount of over-sampling required, neighbours from the k nearest neighbours are randomly chosen. This process is illustrated in Fig. 2, where x_i is the selected point, x_{i1} to x_{i4} are some selected nearest neighbours and r_1 to r_4 the synthetic data points created by the randomized interpolation. The implementation applied in this work uses only one nearest neighbour using the euclidean distance, and balances both classes to the 50% distribution.

Synthetic samples are generated in the following way: Take the difference between the feature vector (sample) under consideration and its nearest neighbour. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features. This approach effectively forces the decision region of the minority class to become more general. An example is detailed in Fig. 3.

In short, its main feature is to form new minority class examples by interpolating between several minority class examples that lie together. Thus, the over-fitting problem is avoided and causes the decision boundaries for the minority class to spread further into the majority class space.

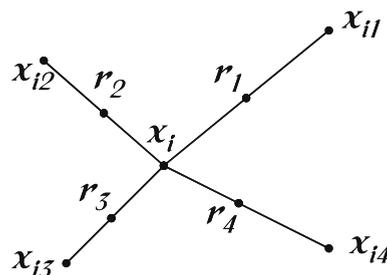


Fig. 2. An illustration of how to create the synthetic data points in the SMOTE algorithm.

Consider a sample (6,4) and let (4,3) be its nearest neighbour.
 (6,4) is the sample for which k-nearest neighbours
 are being identified and (4,3) is one of its k-nearest neighbours.
 Let: $f1_1 = 6$ $f2_1 = 4$, $f2_1 - f1_1 = -2$
 $f1_2 = 4$ $f2_2 = 3$, $f2_2 - f1_2 = -1$
 The new samples will be generated as
 $(f1', f2') = (6,4) + \text{rand}(0-1) * (-2, -1)$
 $\text{rand}(0-1)$ generates a random number between 0 and 1.

Fig. 3. Example of the SMOTE application.

Nevertheless, class clusters may be not well defined in cases where some majority class examples might be invading the minority class space. The opposite can also be true, since interpolating minority class examples can expand the minority class clusters, introducing artificial minority class examples too deeply into the majority class space. Inducing a classifier in such a situation can lead to over-fitting. For this reason we will also consider in this work a hybrid approach, “SMOTE + ENN”, where the Wilson’s Edited Nearest Neighbour Rule [69] is used after the SMOTE application to remove any example from the training set misclassified by its three nearest neighbours.

2.3. Evaluation in imbalanced domains

The measures of the quality of classification are built from a confusion matrix (shown in Table 1) which records correctly and incorrectly recognized examples for each class.

The most used empirical measure, accuracy (1), does not distinguish between the number of correct labels of different classes, which in the ambit of imbalanced problems may lead to erroneous conclusions. For example a classifier that obtains an accuracy of 90% in a data-set with an IR value of 9, might not be accurate if it does not cover correctly any minority class instance.

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \tag{1}$$

Because of this, instead of using accuracy, more correct metrics are considered. Specifically, from Table 1 it is possible to obtain four metrics of performance that measure the classification quality for the positive and negative classes independently:

- **True positive rate** $TP_{rate} = \frac{TP}{TP+FN}$ is the percentage of positive cases correctly classified as belonging to the positive class.
- **True negative rate** $TN_{rate} = \frac{TN}{FP+TN}$ is the percentage of negative cases correctly classified as belonging to the negative class.
- **False positive rate** $FP_{rate} = \frac{FP}{FP+TN}$ is the percentage of negative cases misclassified as belonging to the positive class.
- **False negative rate** $FN_{rate} = \frac{FN}{TP+FN}$ is the percentage of positive cases misclassified as belonging to the negative class.

One appropriate metric that could be used to measure the performance of classification over imbalanced data-sets is the Receiver Operating Characteristic (ROC) graphics [11]. In these graphics, the trade-off between the benefits (TP_{rate}) and costs (FP_{rate}) can be visualized, and acknowledges the fact that the capacity of any classifier cannot increase the number of true positives without also increasing the false positives. The Area Under the ROC Curve (AUC) [41] corresponds to the probability of correctly identifying which of the two stimuli is noise and which is signal plus noise. AUC provides a single-number summary for the performance of learning algorithms.

The way to build the ROC space is to plot on a two-dimensional chart the true positive rate (Y-axis) against the false positive rate (X-axis) as shown in Fig. 4. The points (0,0) and (1,1) are trivial classifiers in which the output class is always predicted as negative and positive respectively, while the point (0,1) represents perfect classification. To compute the AUC we just need to obtain the area of the graphic as:

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \tag{2}$$

3. Fuzzy rule based classification system learning methods

Any classification problem consists of m training patterns $x_p = (x_{p1}, \dots, x_{pn})$, $p = 1, 2, \dots, m$ from M classes where x_{pi} is the i th attribute value ($i = 1, 2, \dots, n$) of the p th training pattern.

In this work we use fuzzy rules of the following form for our FRBCSs:

Table 1

Confusion matrix for a two-class problem.

	Positive prediction	Negative prediction
Positive class	True positive (TP)	False negative (FN)
Negative class	False positive (FP)	True negative (TN)

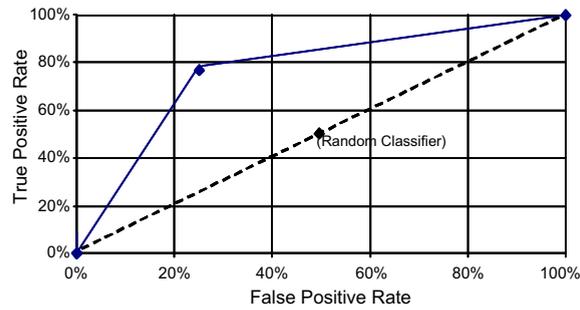


Fig. 4. Example of an ROC plot. Two classifiers are represented: the solid line is a good performing classifier whereas the dashed line represents a random classifier.

$$\text{Rule } R_j : \text{ If } x_1 \text{ is } A_{j1} \text{ and } \dots \text{ and } x_n \text{ is } A_{jn} \text{ then Class} = C_j \text{ with } RW_j \quad (3)$$

where R_j is the label of the j th rule, $x = (x_1, \dots, x_n)$ is an n -dimensional pattern vector, A_{ji} is an antecedent fuzzy set, C_j is a class label, and RW_j is the rule weight [44,74]. We use triangular MFs as antecedent fuzzy sets.

In order to build the RB, we have chosen two fuzzy learning methods: the Chi et al.'s rule generation method [16] and the FH-GBML algorithm [45]. The former has been selected as a classical and simple FRBCS, following the same scheme as our previous works [31–33]. The latter is a recent proposal that presents a good behaviour in standard classification, and our aim is to analyse whether it is accurate for imbalanced data-sets. In the following subsections we will describe these procedures.

3.1. Chi et al. approach

This FRBCSs design method is an extension of the well-known Wang and Mendel method [65] to classification problems. To generate the fuzzy RB, it determines the relationship between the variables of the problem and establishes an association between the space of the features and the space of the classes by means of the following steps:

1. *Establishment of the linguistic partitions.* Once the domain of variation of each feature A_i is determined, the fuzzy partitions are computed.
2. *Generation of a fuzzy rule for each example $x_p = (x_{p1}, \dots, x_{pn}, C_p)$.* To do this it is necessary:
 - 2.1 To compute the matching degree $\mu(x_p)$ of the example to the different fuzzy regions using a conjunction operator (usually modeled with a minimum or product T-norm).
 - 2.2 To assign the example x_p to the fuzzy region with the greatest membership degree.
 - 2.3 To generate a rule for the example, whose antecedent is determined by the selected fuzzy region and whose consequent is the label of class of the example.
 - 2.4 To compute the rule weight.

We must remark that rules with the same antecedent can be generated during the learning process. If they have the same class in the consequent we just remove one of the duplicated rules, but if they have a different class only the rule with the highest weight is kept in the RB.

3.2. Fuzzy Hybrid Genetic Based Machine Learning rule generation algorithm

Different Genetic Fuzzy Systems have been proposed in the specialized literature for designing fuzzy rule based systems in order to avoid the necessity of linguistic knowledge from domain experts [18,37,50,51].

The basis of the algorithm described here [45], consists of a Pittsburgh approach where each rule set is handled as an individual. It also contains a Genetic Cooperative-Competitive Learning (GCCL) approach (an individual represents a unique rule), which is used as a kind of heuristic mutation for partially modifying each rule set, because of its high search ability to efficiently find good fuzzy rules.

The system defines 14 possible linguistic terms for each attribute, as shown in Fig. 5, which correspond to Ruspini's strong fuzzy partitions with two, three, four, and five uniformly distributed triangular-shaped membership functions. Furthermore, the system also uses "don't care" as an additional linguistic term, which indicates that the variable matches any input value with maximum matching degree.

The main steps of this algorithm are described below:

- Step 1: Generate N_{pop} rule sets with N_{rule} fuzzy rules.
- Step 2: Calculate the fitness value of each rule set in the current population.

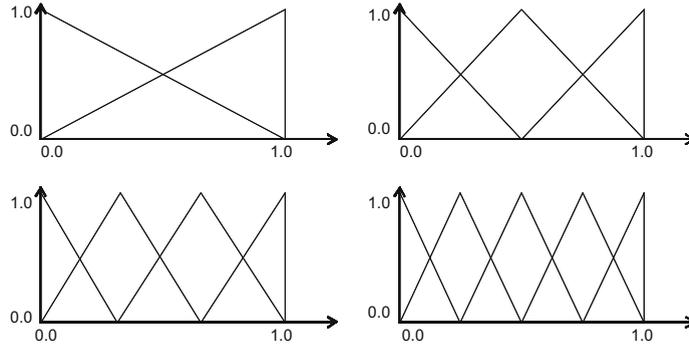


Fig. 5. Four fuzzy partitions for each attribute membership function.

- Step 3: Generate $(N_{pop} - 1)$ rule sets by selection, crossover and mutation in the same manner as the Pittsburgh-style algorithm. Apply a single iteration of the GCCL-style algorithm (i.e., the rule generation and the replacement) to each of the generated rule sets with a pre-specified probability.
- Step 4: Add the best rule set in the current population to the newly generated $(N_{pop} - 1)$ rule sets to form the next population.
- Step 5: Return to Step 2 if the pre-specified stopping condition is not satisfied.

Next, we will describe every step of the algorithm:

- Initialization: N_{rule} training patterns are randomly selected. Then, a fuzzy rule from each of the selected training patterns is generated by choosing probabilistically (as shown in (4)) an antecedent fuzzy set from the 14 candidates B_k ($k = 1, 2, \dots, 14$) (see Fig. 5) for each attribute. Then each antecedent fuzzy set of the generated fuzzy rule is replaced with *don't care* using a pre-specified probability $P_{don't\ care}$.

$$P_{don't\ care}(B_k) = \frac{\mu_{B_k}(x_{pi})}{\sum_{j=1}^{14} \mu_{B_j}(x_{pi})} \quad (4)$$

- Fitness computation: The fitness value of each rule set S_i in the current population is calculated as the number of correctly classified training patterns by S_i . For the GCCL approach the computation follows the same scheme.
- Selection: It is based on binary tournament.
- Crossover: The substring-wise and bit-wise uniform crossover are applied in the Pittsburgh-part. In the case of the GCCL-part only the bit-wise uniform crossover is considered.
- Mutation: Each fuzzy partition of the individuals is randomly replaced with a different fuzzy partition using a pre-specified mutation probability for both approaches.

We must point out that we have used a modification of the fitness function in order to deal directly with imbalanced data. In the case of the Pittsburgh approach, instead of simply using the number of correctly classified patterns, we have computed the AUC measure in order to obtain a good performance for both classes.

4. Genetic tuning of the fuzzy rule based classification systems

The main objective of this work is to improve the performance of FRBCSs in the framework of imbalanced data-sets by means of a tuning approach based on 2-tuples, stressing the positive synergy between this genetic tuning and the FRBCSs in this specific scenario. This methodology consists of refining a previous definition of the DB once the RB has been obtained [4,46,48]. The tuning introduces a variation in the shape of the MFs that improves their global interaction with the main aim of inducing a better cooperation among the rules [20,39]. In this way, the real aim of the tuning is to find the best global configuration of the MFs and not to only find specific MFs in an independent way.

Another possibility, which is out of the scope of this paper, is the tuning of the Inference System parameters [23,32,53]. The use of the appropriate conjunction connectors in the Inference System can improve the fuzzy system behaviour by using parameterised expressions, while maintaining the original interpretability associated with fuzzy systems [5,22].

In the following subsections we will first analyse the significance of the tuning step in fuzzy systems. Then, we will present the tuning approach used in this paper, the lateral tuning approach. Finally, we will describe the evolutionary algorithm used to learn the displacements of the fuzzy partitions.

4.1. Significance of the tuning step

Basic linguistic fuzzy modeling methods are exclusively focused on determining the set of fuzzy rules composing the RB of the model. In these cases, the MFs are usually obtained from expert information (if available) or by a normalization process, and it remains fixed during the RB derivation process.

In the latter case, the fuzzy partitions are not adapted to the context of each variable, because of the limitation of the standard homogenous distribution of the fuzzy labels. Furthermore, the rule extraction method can include some rules with bad performance, and the cooperative behaviour of the rules may not be optimal.

To solve this problem, a post-processing tuning step is used. This step includes a variation in the shape of the MFs that improves their global interaction with the main aim of inducing better cooperation among the rules. In this way, the real aim of the tuning is to find the best global configuration of the MFs and not only to independently find specific MFs.

Classically, the tuning methods refine the three definition parameters that identify triangular MFs associated with the labels comprising the DB [20,26] in order to find its best global configuration (to induce to the best cooperation among the rules). However, in the case of problems with many variables, the dependency among MFs and the dependency among the three definition points leads to tuning models handling very complex search spaces which affect the good performance of the optimization methods [2].

In this work we will apply the 2-tuples based genetic tuning for classification problems, adapting the previous work on the topic [3] in order to obtain good models of FRBCSs to enhance the performance of the initial Knowledge Base (KB).

4.2. Lateral tuning of fuzzy rule based systems

In this approach, a rule representation model based on the linguistic 2-tuples representation [40] is used. This representation allows the lateral displacement of the labels considering only one parameter (slight displacements to the left/right of the original MFs). This involves a simplification of the search space that eases the derivation of optimal models. Furthermore, this process of contextualizing the MFs enables them to achieve a better covering degree while maintaining the original shapes, which results in accuracy improvements without a loss in the interpretability of the fuzzy labels.

The symbolic translation of a linguistic term is a number within the interval $[-0.5, 0.5]$ that expresses the domain of a label when it is moving between its two lateral labels (see Fig. 6). Let us consider a set of labels S representing a fuzzy partition. Formally, we have the pair, (s_i, α_i) , $s_i \in S$, $\alpha_i \in [-0.5, 0.5]$.

As we have said previously, this proposal decreases the tuning problem complexity, since the 3 parameters considered per label are reduced to only 1 symbolic translation parameter. An example is illustrated in Fig. 7 where we show the symbolic translation of a label represented by the pair $(S_2, -0.3)$ together with the lateral displacement of the corresponding MF.

There are two different possible methods to perform the lateral tuning, the most interpretable one, the Global Tuning of the Semantics, and the most accurate one, the Local Tuning of the Rules:

- *Global Tuning of the Semantics (GTS)*: the tuning is applied to the level of linguistic partition. The pair (X_i, label) takes the same tuning value in all the rules where it is considered. For example, X_i is (High, 0.3) will present the same value for those rules in which the couple “ X_i is High” is initially considered. In brief, only one displacement parameter is considered for each label in the DB. Considering this approach, the global interpretability of the final FRBS is maintained. It could be compared to the classical tuning of the DB considering descriptive fuzzy rules [19], i.e., a global collection of fuzzy sets is considered by all the fuzzy rules.
- *Local Tuning of the Rules (LTR)*: the tuning is applied at the rule level. The pair (X_i, label) is tuned in a different way for each rule, based on the quality measures associated with the tuning method. Different displacement parameters are considered for each label in the DB depending on the rule in which this label is considered (one parameter per rule and variable). For example, we could have the pair (X_i, High) in different rules with different displacement parameters:
 Rule k : X_i is (High, 0.3) (more than high)
 Rule j : X_i is (High, -0.2) (a little lower than high)

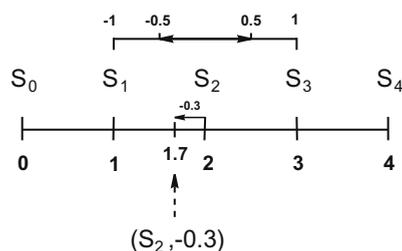


Fig. 6. Symbolic translation of a label.

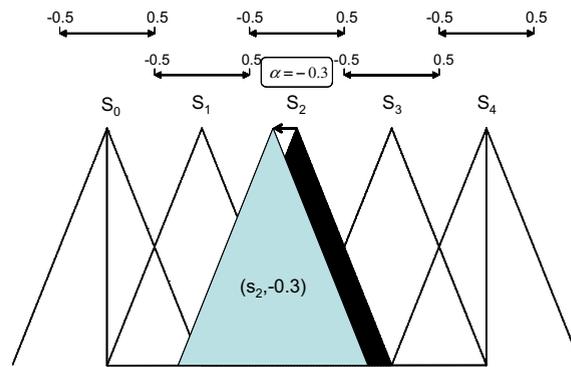


Fig. 7. Lateral displacement of a MF.

In this case, the global interpretability is lost to some degree and, the obtained model should be interpreted from a local point of view. In our experimental study we will apply both approaches in order to determine the behaviour of each one of them for imbalanced data-sets.

4.3. Genetic algorithm for tuning: the CHC algorithm

Genetic Algorithms (GAs) have been widely used to derive fuzzy systems [37]. In this work, we will consider the use of a specific GA to design the proposed learning method, the CHC algorithm [29]. The CHC algorithm is a GA that presents a good trade-off between exploration and exploitation, making it a good choice in problems with complex search spaces. This genetic model makes use of a mechanism of “Selection of Populations”. M parents and their corresponding offspring are put together to select the best M individuals to take part in the next population (with M being the population size).

To provoke diversity in the population, the CHC approach makes use of an incest prevention mechanism and a restarting approach, instead of the well-known mutation operator. This incest prevention mechanism is considered in order to apply the crossover operator, i.e., two parents are recombined if their distance (considering an adequate metric) divided by two is above a predetermined threshold, L . This threshold value is initialized as the maximum possible distance between two individuals divided by four. Following the original CHC scheme, L is decremented by one when there are no new individuals in the population in one generation. When L is below zero the algorithm restarts the population.

The components needed to design this process are explained below. They are: coding scheme, initial gene pool, chromosome evaluation, crossover operator (together with the considered incest prevention) and restarting approach.

1. *Coding Scheme*: As two different types of tuning have been proposed (GTS and LTR), there are two different kinds of coding schemes. In both cases, a real coding is considered, i.e., the real parameters are the GA representation units (genes). Both schemes are presented below:

- GTS: Joint of the parameters of the fuzzy partitions. Let us consider the following number of labels per variable: (m^1, m^2, \dots, m^n) , with n being the number of variables. Then, a chromosome has the form (where each gene is associated with the lateral displacement of the corresponding label in the DB),

$$C_T = (c_{11}, \dots, c_{1m^1}, c_{21}, \dots, c_{2m^2}, \dots, c_{n1}, \dots, c_{nm^n}).$$

An example of a coding scheme considering this approach is shown in Fig. 8a.

- LTR: Joint of the rule parameters. Let us consider that the FRBCS has M rules, (R_1, R_2, \dots, R_M) , with n input variables. Then, the chromosome structure has the following form (where each gene is associated with the lateral displacement of the corresponding label for each rule),

$$C_T = (c_{11}, \dots, c_{1n}, c_{21}, \dots, c_{2n}, \dots, c_{M1}, \dots, c_{Mn}).$$

An example of a coding scheme considering this approach is shown in Fig. 8b.

2. *Chromosome Evaluation*: The fitness function must be in accordance with the framework of imbalanced data-sets. Thus, we will use, as presented in Section 2.3, the AUC measure, defined in (2) as:

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2}$$

3. *Initial Gene Pool*: To make use of the available information, the initial FRBCS is included in the population as an initial solution. This FRBCS can be obtained from an automatic fuzzy rule learning method or from expert knowledge. In this paper, we will use the two fuzzy rule learning algorithms described in Section 3, the Chi et al.’s approach and the FH-GBML algorithm. The initial pool is obtained with the first individual having all genes with the value ‘0.0’, and the remaining individuals generated at random in $[-0.5, 0.5]$.

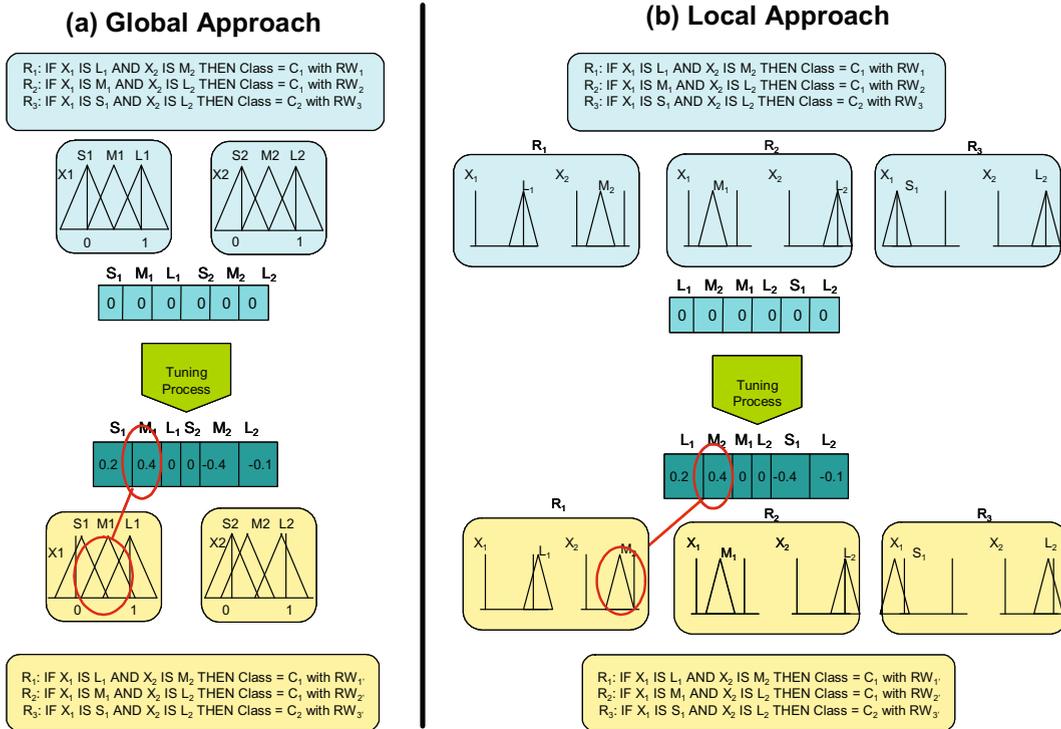


Fig. 8. Example of a coding scheme considering the lateral tuning and rule selection: (a) GTS (global approach) and (b) LTR (local approach).

4. *Crossover Operator*: We consider the Parent Centric BLX (PCBLX) operator [38], which is based on the BLX- α . Fig. 9 depicts the behaviour of these kinds of operators. PCBLX is described as follows. Let us assume that $X = (x_1 \dots x_n)$ and $Y = (y_1 \dots y_n)$, $(x_i, y_i \in [a_i, b_i] \subset \mathfrak{R}, i = 1, \dots, n)$, are two real-coded chromosomes that are going to be crossed. The PCBLX operator generates the two following offspring:

- $O_1 = (o_{11} \dots o_{1n})$, where o_{1i} is a randomly (uniformly) chosen number from the interval $[l_i^1, u_i^1]$, with $l_i^1 = \max\{a_i, x_i - I_i\}$, $u_i^1 = \min\{b_i, x_i + I_i\}$, and $I_i = |x_i - y_i|$.
- $O_2 = (o_{21} \dots o_{2n})$, where o_{2i} is a randomly (uniformly) chosen number from the interval $[l_i^2, u_i^2]$, with $l_i^2 = \max\{a_i, y_i - I_i\}$ and $u_i^2 = \min\{b_i, y_i + I_i\}$.

On the other hand, the incest prevention mechanism will only be considered in order to apply the PCBLX operator. In our case, two parents are crossed if their hamming distance divided by 2 is above a predetermined threshold, L . Since we consider a real coding scheme, we have to transform each gene considering a Gray Code (binary code) with a fixed number of bits per gene (*BITSGENE*), which is determined by the system expert. In this way, the threshold value is initialized as:

$$L = (\#Genes \cdot BITSGENE) / 4.0$$

where $\#Genes$ stands for the total length of the chromosome. Following the original CHC scheme, L is decremented by one (*BITSGENE* in this case) when there are no new individuals in the next generation.

5. *Restarting approach*: Since no mutation is performed, to get away from local optima a restarting mechanism is considered [29] when the threshold value L is lower than zero. In this case, all the chromosomes are generated at random within the interval $[-0.5, 0.5)$. Furthermore, the best global solution found is included in the population to increase the convergence of the algorithm.

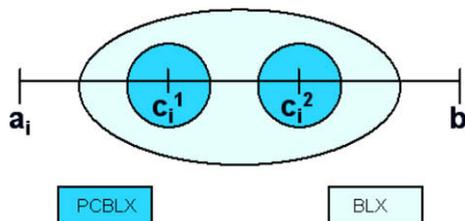


Fig. 9. Scheme of the behaviour of the BLX and PCBLX operators.

We must point out that the RW associated with each fuzzy rule must be recalculated every time the chromosome is decoded (when performing the MF displacement), since the covering degree of the rule may vary.

5. Experimental study

In this paper, we use the IR to distinguish between two classes of imbalanced data-sets: data-sets with a *low imbalance*, when the instances of the positive class are between 10% and 40% of the total instances (IR between 1.5 and 9), and data-sets with a *high imbalance*, where there are no more than 10% of positive instances in the whole data-set compared to the negative ones (IR higher than 9).

We have considered 44 data-sets from the UCI repository [7] with different IR. Table 2 summarizes the data selected in this study and shows, for each data set, the number of examples (#Ex.), number of attributes (#Atts.), class name of each class (minority and majority), class attribute distribution and IR. This table is ordered by the IR, from low to highly imbalanced data-sets.

This study is divided into three parts:

- First, we will use all data-sets in order to analyse the use of a preprocessing step and its synergy with the 2-tuples genetic tuning, contrasting the results of the two FRBCSs learning methods, that is, the Chi et al.'s algorithm and the Ishibuchi and Yamamoto's FH-GBML rule generation with and without tuning when learning directly from the original training set and when the data distribution is balanced artificially.
- Next, we will perform a global comparison among the fuzzy classification methods, and two classical learning algorithms: Ripper, a well-known and accurate rule based method, and C4.5, which has shown a good behaviour in the framework of imbalanced data-sets [55,61,62]. Both methods were run using KEEL software [6], following the recommended parameter values given in the KEEL platform to configure the methods, which also correspond to the settings used in the bibliography of these methods. The FRBCSs will be applied in their basic scheme and using the 2-tuples based genetic tuning. Our aim is to show that the 2-tuples genetic tuning is necessary to improve the behaviour of the simple FRBCS methods, in order to outperform Ripper and the C4.5 decision tree in imbalanced data-sets.
- Finally, we will repeat this analysis in the two groups of imbalanced data-sets previously defined. In this case, we want to study the possible differences between both scenarios in the performance of the FRBCSs against Ripper and C4.5.

In the remainder of this section, we will first present the experimental framework and the parameter configuration for the algorithms selected in this study. Then, we will show our empirical analysis following the outline we described above.

5.1. Experimental set-up

To develop the different experiments we consider a *5-folder cross-validation model*, i.e., five random partitions of data with a 20%, and the combination of 4 of them (80%) as training and the remaining one as test. For each data-set we consider the average results of the five partitions.

Statistical analysis needs to be carried out in order to find significant differences among the results obtained by the studied methods [34]. We consider the use of non-parametric tests, according to the recommendations made in [24,35], where a set of simple, safe and robust non-parametric tests for statistical comparisons of classifiers is presented. For pair-wise comparisons we will use Wilcoxon's signed-ranks test [60,68].

In order to reduce the effect of imbalance, we will use the SMOTE preprocessing method [13] for all our experiments (including the FRBCSs and Ripper). For C4.5 we will use a hybrid approach for SMOTE, SMOTE + ENN [10] that shows a positive synergy when pruning the tree [27]. In both cases, we will consider only the 1-nearest neighbour to generate the synthetic samples, and balancing both classes to the 50% distribution.

We will apply the same configuration for both FRBCS approaches (Chi and FH-GBML), consisting of the product T-norm as conjunction operator, together with the Penalized Certainty Factor approach [44] for the rule weight and FRM of the winning rule. We have selected this FRBCS model as it achieved a good performance in our former studies on imbalanced data-sets [33]. Because it is not clear what level of granularity must be selected for the Chi FRBCS, we will use both three and five labels per variable.

In the case of the Ishibuchi and Yamamoto's FH-GBML method, we consider the following values for the parameters:

- Number of fuzzy rules: $5 \cdot d$ rules.
- Number of rule sets: 200 rule sets.
- Crossover probability: 0.9.
- Mutation probability: $1/d$.
- Number of replaced rules: All rules except the best-one (Pittsburgh-part, elitist approach), number of rules/5 (GCCL-part).
- Total number of generations: 1000 generations.
- Don't care probability: 0.5.

Table 2
Summary description for imbalanced data-sets.

Data-set	# Ex.	# Atts.	Class(min., maj.)	% Class(min.; maj.)	IR
<i>Data-sets with Low Imbalance (IR 1.5–9)</i>					
Glass1	214	9	(build-win-non_float-proc; remainder)	(35.51,64.49)	1.82
Ecoli0vs1	220	7	(im; cp)	(35.00,65.00)	1.86
Wisconsin	683	9	(malignant; benign)	(35.00,65.00)	1.86
Pima	768	8	(tested-positive; tested-negative)	(34.84,66.16)	1.90
Iris0	150	4	(Iris-Setosa; remainder)	(33.33,66.67)	2.00
Glass0	214	9	(build-win-float-proc; remainder)	(32.71,67.29)	2.06
Yeast1	1484	8	(nuc; remainder)	(28.91,71.09)	2.46
Vehicle1	846	18	(Saab; remainder)	(28.37,71.63)	2.52
Vehicle2	846	18	(Bus; remainder)	(28.37,71.63)	2.52
Vehicle3	846	18	(Opel; remainder)	(28.37,71.63)	2.52
Haberman	306	3	(Die; Survive)	(27.42,73.58)	2.68
Glass0123vs456	214	9	(non-window glass; remainder)	(23.83,76.17)	3.19
Vehicle0	846	18	(Van; remainder)	(23.64,76.36)	3.23
Ecoli1	336	7	(im; remainder)	(22.92,77.08)	3.36
New-thyroid2	215	5	(hypo; remainder)	(16.89,83.11)	4.92
New-thyroid1	215	5	(hyper; remainder)	(16.28,83.72)	5.14
Ecoli2	336	7	(pp; remainder)	(15.48,84.52)	5.46
Segment0	2308	19	(brickface; remainder)	(14.26,85.74)	6.01
Glass6	214	9	(headlamps; remainder)	(13.55,86.45)	6.38
Yeast3	1484	8	(me3; remainder)	(10.98,89.02)	8.11
Ecoli3	336	7	(imU; remainder)	(10.88,89.12)	8.19
Page-blocks0	5472	10	(remainder; text)	(10.23,89.77)	8.77
<i>Data-sets with High Imbalance (IR higher than 9)</i>					
Yeast2vs4	514	8	(cyt; me2)	(9.92,90.08)	9.08
Yeast05679vs4	528	8	(me2; mit,me3,exc,vac,erl)	(9.66,90.34)	9.35
Vowel0	988	13	(hid; remainder)	(9.01,90.99)	10.10
Glass016vs2	192	9	(ve-win-float-proc; build-win-float-proc, build-win-non_float-proc,headlamps)	(8.89,91.11)	10.29
Glass2	214	9	(Ve-win-float-proc; remainder)	(8.78,91.22)	10.39
Ecoli4	336	7	(om; remainder)	(6.74,93.26)	13.84
Yeast1vs7	459	8	(nuc; vac)	(6.72,93.28)	13.87
Shuttle0vs4	1829	9	(Rad Flow; Bypass)	(6.72,93.28)	13.87
Glass4	214	9	(containers; remainder)	(6.07,93.93)	15.47
Page-blocks13vs2	472	10	(graphic; horiz.line,picture)	(5.93,94.07)	15.85
Abalone9vs18	731	8	(18; 9)	(5.65,94.25)	16.68
Glass016vs5	184	9	(tableware; build-win-float-proc, build-win-non_float-proc,headlamps)	(4.89,95.11)	19.44
Shuttle2vs4	129	9	(Fpv Open; Bypass)	(4.65,95.35)	20.5
Yeast1458vs7	693	8	(vac; nuc,me2,me3,pox)	(4.33,95.67)	22.10
Glass5	214	9	(tableware; remainder)	(4.20,95.80)	22.81
Yeast2vs8	482	8	(pox; cyt)	(4.15,95.85)	23.10
Yeast4	1484	8	(me2; remainder)	(3.43,96.57)	28.41
Yeast1289vs7	947	8	(vac; nuc,cyt,pox,erl)	(3.17,96.83)	30.56
Yeast5	1484	8	(me1; remainder)	(2.96,97.04)	32.78
Ecoli0137vs26	281	7	(pp,imL; cp,im,imU,imS)	(2.49,97.51)	39.15
Yeast6	1484	8	(exc; remainder)	(2.49,97.51)	39.15
Abalone19	4174	8	(19; remainder)	(0.77,99.23)	128.87

- Probability of the application of the GCCL iteration: 0.5. where d stands for the dimensionality of the problem (number of variables).

Finally, we indicate the values that have been considered for the parameters of the genetic tuning:

- Population size: 50 individuals.
- Number of evaluations: $5000 \cdot d$.
- Bits per gene for the Gray codification (for incest prevention): 30 bits.

5.2. Study of the use of preprocessing on fuzzy rule based classification systems with 2-tuples genetic tuning

In this first part of our study, we will perform our analysis without taking into account the IR of the data-sets. Our aim here is to analyse two different aspects:

1. The improvement obtained for FRBCs by means of the 2-tuples genetic tuning when it is directly applied to the original imbalanced data-sets.

2. Whether the use of preprocessing supposes a positive synergy with the genetic tuning and enables the achievement of more accurate results.

Table 3 shows the global average results for the FRBCS algorithms. By rows, we can observe three blocks of results, the first two ones are related to the Chi et al.'s method (with three and five labels per variable) and the last one is related to the FH-GBML algorithm. This table is also divided by columns into two blocks, on the left-hand side we show the results for the original data-sets whereas on the right-hand side we show the results when we apply a preprocessing step using the SMOTE algorithm. We stress in boldface the best results for each block, that is, for each algorithm and for the original data-sets and preprocessing respectively. The complete table of results for all data-sets is shown in the appendix of this work.

From this table of results we can observe that the highest average value always corresponds to the tuning approach for all FRBCSs in both cases, which suggests the goodness of this technique. We will focus in this part of the study on the results without preprocessing, and thus in Table 4 a Wilcoxon test is shown in which we detect significant differences in favour of the 2-tuples tuning approach for the three methods compared, which supports our previous conclusion.

We can also observe in Table 3 that, for the Chi et al.'s method without preprocessing (both with three and five labels), the higher training results are associated to GTS rather than LTR, which includes more parameters for the tuning of the fuzzy system. This can be due to the fact that the genetic search procedure of the LTR approach falls more easily onto a local optima because we have a limited quality for the RB generated by the Chi et al.'s method.

Regarding the use of preprocessing, the improvement in the performance of the results in the case of the application of SMOTE is clearly shown, with an increase from 5 to 10 points for the different approaches. The comparative graph in Fig. 10 illustrates the differences in proportion between the results for AUC in the test partitions with the original data-sets and with preprocessing. Consequently, we will focus only on the results with SMOTE preprocessing for the remainder of this section.

Finally, we observe that there is a higher difference between the performance in training and test for the Chi et al.'s method in the case of the application of SMOTE both for GTS and LTR. This behaviour is caused by a better set of rules obtained from the preprocessed training set, which enable a better tuning that results on a higher precision the training partitions, but which may also cause a slight overtraining. Specifically, there is a clear over-fitting for the Chi et al.'s method with five labels per variable, but this is due to the increasing of the granularity of the fuzzy partitions and the generation of more specific rules for the training data. Nevertheless, we observe that for the FH-GBML algorithm there are neither a change in the difference of performance between train and test without and with preprocessing nor an accentuated overtraining such in the case of the Chi et al.'s method. This different behaviour can be explained regarding the compactness and quality of the RB extracted by this method, which results on a good generalisation capability both when using the original data-sets and when applying SMOTE preprocessing.

Table 3

Table of results for FRBCSs (simple approach and with 2-tuples genetic tuning) for all data-sets. Original data-sets (none) and preprocessing (SMOTE).

Algorithm	None		SMOTE	
	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
Chi-3	68.18 ± 1.71	65.43 ± 4.50	85.56 ± 1.67	81.33 ± 6.57
Chi-3-GTS	80.42 ± 2.46	74.38 ± 5.38	92.89 ± 1.02	83.97 ± 6.63
Chi-3-LTR	76.91 ± 2.23	70.30 ± 4.85	94.63 ± 1.03	84.39 ± 6.69
Chi-5	80.75 ± 1.63	70.61 ± 6.04	90.58 ± 0.96	80.10 ± 6.64
Chi-5-GTS	88.65 ± 1.28	74.06 ± 6.29	94.79 ± 1.00	79.98 ± 6.94
Chi-5-LTR	87.05 ± 1.40	72.78 ± 6.31	96.59 ± 0.76	81.34 ± 6.73
FH-GBML	78.25 ± 2.78	73.12 ± 6.00	89.64 ± 1.42	83.80 ± 6.46
FH-GBML-GTS	84.78 ± 3.42	77.40 ± 6.14	92.56 ± 1.16	84.56 ± 6.40
FH-GBML-LTR	85.44 ± 3.70	77.55 ± 5.69	92.81 ± 1.41	84.65 ± 6.02

Table 4

Wilcoxon test to compare the simple FRBCS approaches (R^+) with the use of 2-tuples tuning (R^-) with the original data-sets.

Comparison	R^+	R^-	p-Value
Chi-3 vs. Chi-3-GTS	52.0	938.0	0.000
Chi-3 vs. Chi-3-LTR	96.0	894.0	0.000
Chi-5 vs. Chi-5-GTS	282.5	707.5	0.008
Chi-5 vs. Chi-5-LTR	207.0	783.0	0.002
FH-GBML vs. FH-GBML-GTS	94.5	895.5	0.000
FH-GBML vs. FH-GBML-LTR	112.5	877.5	0.000

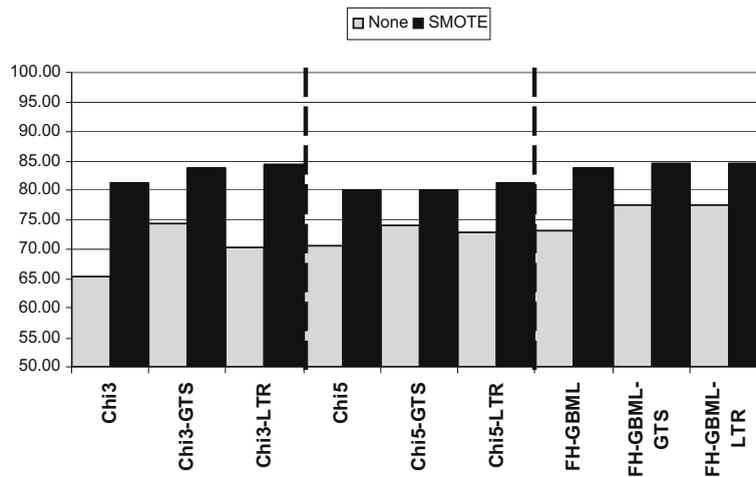


Fig. 10. Comparative graph between the use of the original data-sets and preprocessing for the FRBCSs, with and without 2-tuples genetic tuning. The height of the bars represents the average performance with AUC in the test partitions.

5.3. Global analysis of the 2-tuples based genetic tuning on fuzzy rule based classification systems with preprocessing

This study is divided into two parts: first, we will present a global comparison between the Chi et al.'s rule generation method [16] and the Ishibuchi and Yamamoto's FH-GBML [45] by contrasting them in their basic approach and using the 2-tuples based genetic tuning in both the global and local approaches. Then, we will include Ripper and C4.5 in our statistical study to analyse the differences when comparing the FRBCS approaches with and without tuning against Ripper and C4.5. As we have stated in the previous section, we have included the complete tables of results for all the implemented algorithms in the appendix of this work. These results will be analysed next.

For the FRBCSs analysis we must select which granularity is preferred for the Chi method, whether three or five labels. For this purpose, Table 5 shows the experimental results, where we show in columns the Chi et al.'s algorithm with three and five labels, noted as Chi-3 and Chi-5, respectively. In addition, there are three different results for each method: the first row contains the results when applying the basic scheme (Base) and the second and third rows contain the results for the global and local 2-tuples based genetic tuning, named GTS and LTR.

Table 6 presents a Wilcoxon test where we compare the results for each approach (with the two types of genetic tuning) using the two different numbers of fuzzy partitions. In this test, R^+ corresponds to the sum of ranks for the data-sets in which the first algorithm outperformed the second, and R^- the sum of ranks for the opposite.

The main conclusion extracted from this table is that when we choose five labels per variable, we get a high over-fitting for the 2-tuples based genetic tuning and, in this case, the choice of a lower level of granularity allows better results to be achieved.

Next, we analyse the behaviour of the 2-tuples genetic tuning over all imbalanced data-sets. For this purpose, we present in Table 7 the results of the Chi et al.'s approach (with three labels per variable) and the FH-GBML algorithm, to study the improvement achieved in the case of the post-processing step. Now, we will include in this table the results obtained with Ripper and the C4.5 decision tree, since we will also compare the performance of the 2-tuples genetic tuning with these well-known algorithms. The complete results table, with the performance obtained in each data-set in the test partitions, is shown in the next subsection.

Table 5

Results table for Chi in all imbalanced data-sets.

Approach	Chi-3		Chi-5	
	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
Base	85.56 ± 1.67	81.33 ± 6.57	90.58 ± 0.96	80.10 ± 6.64
GTS	92.89 ± 1.02	83.97 ± 6.63	94.79 ± 1.00	79.98 ± 6.94
LTR	94.63 ± 1.03	84.39 ± 6.69	96.59 ± 0.76	81.34 ± 6.73

Table 6

Wilcoxon test to compare Chi using different granularity levels. R^+ corresponds to three labels and R^- to five labels.

Comparison	R^+	R^-	p-Value
Chi-3-GTS vs. Chi-5-GTS	878.5	111.5	0.000
Chi-3-LTR vs. Chi-5-LTR	840.0	150.0	0.000

Table 7

Table of results for FRBCSs (simple approach and with 2-tuples genetic tuning) Ripper and C4.5 for all data-sets. SMOTE preprocessing is applied to FRBCSs and Ripper. SMOTE + ENN is applied for C4.5.

Algorithm	AUC_{Tr}	AUC_{Tst}
Chi-3	85.56 ± 1.67	81.33 ± 6.57
Chi-3-GTS	92.89 ± 1.02	83.97 ± 6.63
Chi-3-LTR	94.63 ± 1.03	84.39 ± 6.69
FH-GBML	89.64 ± 1.42	83.80 ± 6.46
FH-GBML-GTS	92.56 ± 1.16	84.56 ± 6.40
FH-GBML-LTR	92.81 ± 1.41	84.65 ± 6.02
Ripper	95.98 ± 0.91	83.54 ± 6.15
C4.5	95.35 ± 1.20	83.75 ± 5.52

The first analysis is shown in Table 8, in which a Wilcoxon test help us to determine that in both cases (Chi and FH-GBML) the 2-tuples tuning improves the behaviour of the simple KB, both in the global and local approaches. Therefore, we emphasize the goodness of the 2-tuples methodology for the tuning of the MF in imbalanced data-sets, both for the whole rule set and for each fuzzy rule.

Our intention is to show that the use of the 2-tuples genetic tuning enables the FRBCSs to become competitive and even outperform the Ripper algorithm and the C4.5 decision tree. Thus, we show in Table 9 a comparison among Ripper, C4.5 and the different FRBCSs approaches. We can observe in this table that, in the case of the simple FRBCSs, the Chi et al.'s algorithm is significantly worse than Ripper and C4.5, whereas for the FH-GBML the null hypothesis of equality cannot be rejected. Nevertheless, when the 2-tuples genetic tuning is applied for the FRBCSs, we always obtained the best ranking (except in the case of Chi-3-GTS), and our approach is statistically better than Ripper for FH-GBML-GTS and outperforms Ripper and C4.5 for FH-GBML-LTR with a low p -value.

Finally, we show in Table 10 the average number of rules obtained by each one of the algorithms used in this paper, which is updated with the total number of rules extracted for every single data-set in the appendix of this work.

We can observe that, whereas the highest complexity corresponds to the Chi et al.'s method, the FH-GBML presents a similar number of rules to the algorithms of comparison Ripper and C4.5. In fact, its number of rules is lower than that of C4.5

Table 8

Wilcoxon test to compare the simple FRBCS approaches (R^+) with the use of 2-tuples tuning (R^-) with preprocessing.

Comparison	R^+	R^-	p -Value
Chi-3 vs. Chi-3-GTS	176.5	813.5	0.000
Chi-3 vs. Chi-3-LTR	164.5	825.5	0.000
FH-GBML vs. FH-GBML-GTS	288.5	701.5	0.018
FH-GBML vs. FH-GBML-LTR	241.0	749.0	0.003

Table 9

Wilcoxon test to compare the performance of Ripper and C4.5 (R^+) with the FRBCSs with and without tuning (R^-) in all imbalanced data-sets.

Comparison	R^+	R^-	p -Value
Ripper vs. Chi-3	695	295	0.020
Ripper vs. Chi-3-GTS	444.5	545.5	0.570
Ripper vs. Chi-3-LTR	390	600	0.220
Ripper vs. FH-GBML	474	516	0.804
Ripper vs. FH-GBML-GTS	352	638	0.095
Ripper vs. FH-GBML-LTR	347	643	0.084
C4.5 vs. Chi-3	701	289	0.016
C4.5 vs. Chi-3-GTS	528	462	0.700
C4.5 vs. Chi-3-LTR	415	575	0.351
C4.5 vs. FH-GBML	494	496	0.933
C4.5 vs. FH-GBML-GTS	417	573	0.363
C4.5 vs. FH-GBML-LTR	346	644	0.082

Table 10

Average number of rules table for FRBCSs, Ripper and C4.5.

Algorithm	Number of rules
Chi-3	96.69 ± 3.47
FH-GBML	19.65 ± 5.71
Ripper	14.40 ± 1.91
C4.5	26.05 ± 3.66

and it is not comparable with Ripper since this algorithm manages a decision list, which has a completely different philosophy to the set of rules of the remaining approaches of this work. We conclude that the FH-GBML is a good mechanism for imbalanced data-sets since it has a low complexity (number of rules) and achieves a good performance, which is significantly improved by the application of the 2-tuples genetic tuning approach, as was shown during this empirical study. Furthermore, rules with fuzzy labels are usually easier to manage by an expert rather than interval rules, which implies an additional advantage for the FRBCSs.

5.4. Analysis of the 2-tuples based genetic tuning on fuzzy rule based classification systems with preprocessing according to the imbalance ratio

In the last part of our study, we want to analyse the behaviour of the 2-tuples genetic tuning in the two scenarios of imbalanced data-sets proposed in this paper, i.e. data-sets with a low and a high imbalance.

Table 11

Detailed table of results for Chi et al. and FH-GBML with 2-tuples based genetic tuning. Including the results for Ripper and C4.5.

Data-set	Chi-3	Chi-3-GTS	Chi-3-LTR	FH-GBML	FH-GBML-GTS	FH-GBML-LTR	Ripper	C4.5
<i>Data-sets with low imbalance ($1.5 \leq IR < 9$)</i>								
Glass1	65.53 ± 6.93	66.20 ± 6.63	74.47 ± 4.62	72.68 ± 2.79	76.34 ± 6.22	77.84 ± 4.26	73.26 ± 9.27	75.77 ± 3.79
Ecoli0vs1	92.71 ± 5.52	95.24 ± 3.60	93.93 ± 4.34	97.02 ± 1.80	97.02 ± 1.80	95.98 ± 2.54	98.32 ± 2.38	97.96 ± 2.19
Wisconsin	89.19 ± 1.98	93.13 ± 0.97	94.96 ± 1.04	96.59 ± 0.72	96.82 ± 0.96	96.84 ± 1.51	96.36 ± 1.55	95.45 ± 1.99
Pima	67.66 ± 5.20	72.82 ± 5.32	72.35 ± 3.35	73.40 ± 3.54	74.32 ± 3.91	74.57 ± 3.75	70.10 ± 5.46	71.45 ± 3.88
Iris0	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	98.00 ± 4.47	99.00 ± 2.24
Glass0	69.74 ± 2.96	81.48 ± 4.05	81.35 ± 5.29	81.38 ± 3.66	80.64 ± 3.01	83.84 ± 2.89	79.05 ± 7.38	78.56 ± 2.34
Yeast2	69.44 ± 1.74	71.24 ± 2.69	71.33 ± 1.82	69.86 ± 4.04	70.83 ± 3.63	70.69 ± 3.64	69.67 ± 2.69	71.09 ± 2.86
Vehicle2	85.55 ± 3.35	92.13 ± 2.07	92.10 ± 1.48	89.85 ± 2.74	92.35 ± 4.44	93.68 ± 2.76	94.69 ± 2.88	94.92 ± 1.61
Vehicle1	71.40 ± 4.67	74.55 ± 2.90	71.64 ± 3.38	70.01 ± 4.31	72.05 ± 4.27	73.14 ± 6.54	75.09 ± 3.06	70.30 ± 2.90
Vehicle3	69.51 ± 5.07	72.33 ± 3.74	70.05 ± 3.99	70.62 ± 3.30	73.41 ± 4.10	73.09 ± 3.72	71.98 ± 3.47	74.44 ± 1.08
Haberman	60.60 ± 4.38	59.42 ± 4.38	61.71 ± 4.14	62.06 ± 4.65	63.23 ± 5.28	61.93 ± 3.36	56.41 ± 6.19	63.09 ± 4.07
Glass0123vs456	86.42 ± 2.59	90.02 ± 5.29	90.41 ± 2.81	93.14 ± 3.92	92.84 ± 2.80	92.84 ± 2.80	89.69 ± 5.24	90.32 ± 3.12
Vehicle0	86.96 ± 2.99	87.45 ± 1.82	89.77 ± 1.11	88.87 ± 3.29	92.08 ± 2.13	92.88 ± 1.19	93.51 ± 3.95	91.18 ± 2.63
Ecoli1	85.88 ± 9.28	85.29 ± 10.89	84.02 ± 8.85	87.05 ± 6.39	88.82 ± 3.64	90.24 ± 3.33	86.07 ± 4.27	77.55 ± 7.87
New-Thyroid2	90.60 ± 9.71	97.18 ± 4.02	96.03 ± 5.52	98.89 ± 1.16	96.59 ± 6.88	99.44 ± 0.76	97.18 ± 4.13	96.59 ± 4.78
New-Thyroid1	88.33 ± 7.05	99.17 ± 0.76	96.31 ± 3.98	94.33 ± 5.64	94.60 ± 5.85	94.01 ± 7.59	95.20 ± 4.88	98.02 ± 3.71
Ecoli2	88.26 ± 5.45	87.53 ± 5.30	88.53 ± 6.98	86.48 ± 6.29	88.59 ± 5.88	88.26 ± 5.10	86.03 ± 4.69	91.62 ± 4.85
Segment0	95.07 ± 0.43	98.83 ± 0.69	99.12 ± 0.83	97.98 ± 1.06	98.71 ± 0.83	99.04 ± 0.25	98.66 ± 0.83	99.27 ± 0.60
Glass6	84.69 ± 9.16	85.77 ± 6.64	83.84 ± 6.39	84.42 ± 10.51	82.22 ± 11.50	85.09 ± 10.11	88.69 ± 10.97	84.50 ± 7.50
Yeast3	90.22 ± 4.18	91.63 ± 2.91	91.77 ± 2.74	92.44 ± 3.05	91.45 ± 2.72	92.06 ± 3.06	89.11 ± 0.83	88.76 ± 3.46
Ecoli3	87.84 ± 4.12	91.50 ± 5.15	90.47 ± 5.38	90.27 ± 2.63	92.25 ± 4.05	90.90 ± 4.26	85.02 ± 9.77	89.21 ± 7.15
Page-Blocks0	81.40 ± 3.54	89.17 ± 1.51	86.04 ± 8.55	90.67 ± 0.96	92.76 ± 0.90	93.24 ± 1.83	94.57 ± 0.86	94.85 ± 1.51
Mean	82.14 ± 4.56	85.55 ± 3.70	85.46 ± 3.94	85.82 ± 3.48	86.72 ± 3.85	87.25 ± 3.42	85.76 ± 4.51	86.09 ± 3.46
<i>Data-Sets with High Imbalance ($IR \leq 9$)</i>								
Yeast2vs4	87.36 ± 5.16	87.88 ± 2.35	89.42 ± 2.94	92.50 ± 2.26	93.35 ± 2.90	93.25 ± 3.24	87.03 ± 8.81	85.88 ± 8.78
Yeast05679vs4	79.17 ± 5.66	78.65 ± 10.24	78.15 ± 6.53	79.38 ± 5.58	79.79 ± 3.70	77.47 ± 6.10	74.08 ± 5.21	76.02 ± 9.36
Vowel0	98.39 ± 0.60	98.05 ± 2.88	99.39 ± 1.22	92.94 ± 3.85	96.55 ± 2.37	96.05 ± 1.58	95.78 ± 4.04	94.94 ± 4.95
Glass016vs2	54.17 ± 6.82	56.24 ± 17.35	68.36 ± 11.94	59.60 ± 8.44	63.60 ± 6.73	59.67 ± 6.59	63.71 ± 8.51	60.62 ± 12.66
Glass2	55.30 ± 14.48	71.31 ± 14.50	76.12 ± 13.56	67.29 ± 16.24	67.98 ± 17.79	59.65 ± 10.58	62.17 ± 14.32	54.24 ± 14.01
Ecoli4	91.51 ± 7.21	92.62 ± 7.01	91.71 ± 6.23	88.10 ± 5.80	88.11 ± 6.12	87.94 ± 6.07	88.42 ± 10.46	83.10 ± 9.90
shuttle0vs4	99.12 ± 1.14	99.12 ± 1.14	99.15 ± 1.16	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	99.97 ± 0.07	99.97 ± 0.07
yeastB1vs7	80.63 ± 6.61	77.61 ± 10.17	76.37 ± 11.90	74.06 ± 10.42	70.67 ± 12.35	74.24 ± 12.12	65.83 ± 9.03	70.03 ± 1.46
Glass4	85.70 ± 12.92	89.92 ± 10.91	92.51 ± 10.52	87.67 ± 9.62	85.33 ± 10.29	86.33 ± 10.51	89.67 ± 11.46	85.08 ± 9.35
Page-Blocks13vs4	92.05 ± 4.73	94.76 ± 5.00	97.44 ± 4.18	98.88 ± 0.79	99.32 ± 0.62	99.10 ± 0.50	98.88 ± 0.89	99.55 ± 0.47
Abalone19	63.94 ± 9.32	66.32 ± 13.90	61.58 ± 11.89	64.89 ± 18.63	63.17 ± 14.55	67.64 ± 11.77	53.03 ± 4.54	52.02 ± 4.41
Glass016vs5	79.71 ± 23.29	85.71 ± 21.62	86.29 ± 21.89	86.86 ± 24.65	87.71 ± 21.91	87.43 ± 21.75	94.86 ± 2.17	81.29 ± 24.44
shuttle2vs4	90.78 ± 7.80	98.38 ± 1.68	98.38 ± 2.61	97.95 ± 2.92	97.95 ± 2.92	97.95 ± 2.92	99.58 ± 0.93	99.17 ± 1.86
Yeast1458vs7	64.65 ± 5.92	69.31 ± 6.11	65.20 ± 9.03	58.94 ± 11.30	63.71 ± 7.86	64.53 ± 9.19	63.15 ± 9.62	53.67 ± 2.09
Glass5	83.17 ± 11.12	82.56 ± 20.73	88.29 ± 21.46	88.78 ± 21.69	88.78 ± 21.72	89.27 ± 22.64	93.29 ± 11.19	88.29 ± 13.31
Yeast2vs8	77.28 ± 10.36	76.09 ± 10.10	79.35 ± 11.40	76.11 ± 11.36	77.30 ± 13.45	75.55 ± 10.31	84.57 ± 12.34	80.66 ± 11.22
Yeast4	83.15 ± 2.96	82.42 ± 4.58	78.62 ± 7.87	79.95 ± 5.99	78.70 ± 5.43	79.86 ± 7.01	76.42 ± 9.05	70.04 ± 5.65
Yeast1289vs7	77.12 ± 6.50	70.07 ± 8.70	71.22 ± 6.30	67.13 ± 6.86	71.44 ± 4.37	70.49 ± 4.84	73.65 ± 4.50	68.32 ± 6.16
Yeast5	93.58 ± 5.11	95.90 ± 2.61	95.10 ± 2.74	96.74 ± 2.47	95.87 ± 3.20	95.87 ± 3.28	93.23 ± 5.39	92.33 ± 4.72
Yeast6	88.09 ± 9.82	84.80 ± 7.38	85.11 ± 7.84	86.76 ± 8.91	85.88 ± 7.81	86.26 ± 7.66	79.22 ± 8.87	82.80 ± 12.77
Ecoli0137vs26	81.90 ± 20.49	81.90 ± 20.79	83.18 ± 21.18	81.36 ± 20.13	81.00 ± 19.33	82.27 ± 20.40	82.08 ± 20.99	81.36 ± 21.68
Abalone9-18	64.70 ± 10.73	73.10 ± 10.78	71.98 ± 13.39	73.14 ± 9.88	76.65 ± 11.31	74.28 ± 10.65	70.63 ± 8.81	62.15 ± 4.96
Mean	80.52 ± 8.58	82.40 ± 9.57	83.31 ± 9.45	81.77 ± 9.45	82.40 ± 8.94	82.05 ± 8.62	81.33 ± 7.78	78.25 ± 8.38
<i>All data-sets</i>								
Mean	81.33 ± 6.57	83.97 ± 6.63	84.39 ± 6.69	83.80 ± 6.46	84.56 ± 6.40	84.65 ± 6.02	83.54 ± 6.15	82.17 ± 5.92

Following this idea, Table 11 shows the results in test for the FRBCS algorithms with the GTS and LTR tuning approaches, and for the Ripper algorithm and the C4.5 decision tree. This table is divided by the IR, the first part corresponds to data-sets with a low imbalance and the second part to data-sets with a high imbalance. The best global result for test is stressed in boldface in each case. Furthermore, in Table 12 we show the average results for the two groups of imbalanced data-sets considered. Please refer to the appendix of this work where we show both training and test results for every data-set.

We apply a Wilcoxon test in order to compare Ripper (Table 13) and C4.5 (Table 14) with the FRBCSs (with and without 2-tuples based genetic tuning) and thus, to analyse whether the difference in the average results for the AUC measure is enough to determine statistically that our approach performs better than the selected algorithms of contrast in each one of the imbalanced scenarios.

For data-sets with a low imbalance, the Chi et al.'s approach obtains a low ranking in comparison with Ripper and C4.5, whereas the FH-GBML has a similar behaviour to those methods. When applying the 2-tuples genetic tuning, we can observe

Table 12

Average table of results for FRBCSs, Ripper and C4.5 for the different degrees of imbalance.

Algorithm	Low imbalance		High imbalance	
	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
Chi-3	86.03 ± 1.22	82.14 ± 4.56	85.09 ± 2.12	80.52 ± 8.58
Chi-3-GTS	92.64 ± 0.75	85.55 ± 3.70	93.15 ± 1.29	82.40 ± 9.57
Chi-3-LTR	94.07 ± 0.89	85.46 ± 3.94	95.20 ± 1.17	83.31 ± 9.45
FH-GBML	89.40 ± 0.98	85.82 ± 3.48	89.89 ± 1.85	81.77 ± 9.45
FH-GBML-GTS	91.90 ± 1.04	86.72 ± 3.85	93.21 ± 1.29	82.40 ± 8.94
FH-GBML-LTR	92.38 ± 1.18	87.25 ± 3.42	93.25 ± 1.63	82.05 ± 8.62
Ripper	94.15 ± 1.06	85.76 ± 4.51	97.81 ± 0.77	81.33 ± 7.78
C4.5	94.01 ± 1.00	86.18 ± 3.39	96.69 ± 1.40	81.31 ± 7.65

Table 13Wilcoxon test to compare the performance of Ripper (R^+) with the FRBCS approaches with and without tuning (R^-) in data-sets with a low and a high imbalance.

Comparison	R^+	R^-	p -Value
<i>Data-sets with low imbalance</i>			
Ripper vs. Chi-3	220	33	0.002
Ripper vs. Chi-3-GTS	131.5	121.5	0.821
Ripper vs. Chi-3-LTR	134	119	0.808
Ripper vs. FH-GBML	122	131	0.884
Ripper vs. FH-GBML-GTS	79	174	0.123
Ripper vs. FH-GBML-LTR	60	193	0.031
<i>Data-sets with high imbalance</i>			
Ripper vs. Chi-3	142	111	0.615
Ripper vs. Chi-3-GTS	100	153	0.390
Ripper vs. Chi-3-LTR	72	181	0.077
Ripper vs. FH-GBML	119.5	133.5	0.794
Ripper vs. FH-GBML-GTS	95	158	0.306
Ripper vs. FH-GBML-LTR	112	141	0.638

Table 14Wilcoxon test to compare the performance of C4.5 (R^+) with the FRBCS approaches with and without tuning (R^-) in data-sets with a low and a high imbalance.

Comparison	R^+	R^-	p -Value
<i>Data-sets with low imbalance</i>			
C4.5 vs. Chi-3	224	29	0.002
C4.5 vs. Chi-3-GTS	165	88	0.211
C4.5 vs. Chi-3-LTR	154	99	0.372
C4.5 vs. FH-GBML	151.5	101.5	0.394
C4.5 vs. FH-GBML-GTS	103	150	0.445
C4.5 vs. FH-GBML-LTR	82	171	0.149
<i>Data-sets with high imbalance</i>			
C4.5 vs. Chi-3	134	119	0.808
C4.5 vs. Chi-3-GTS	108	145	0.548
C4.5 vs. Chi-3-LTR	61	192	0.033
C4.5 vs. FH-GBML	100.5	152.5	0.455
C4.5 vs. FH-GBML-GTS	109	144	0.570
C4.5 vs. FH-GBML-LTR	95	158	0.306

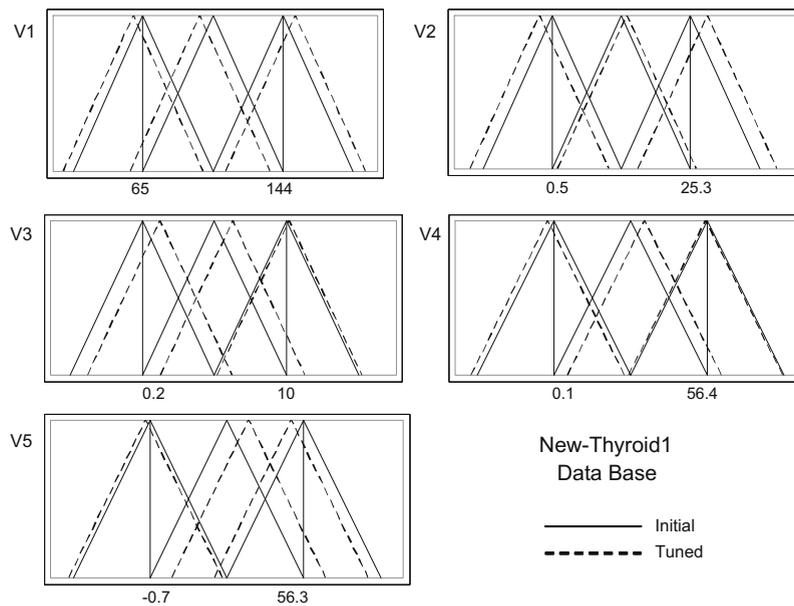


Fig. 11. Initial and tuned DB of a model obtained with GTS in the new-thyroid1 data-set.

an improvement in the performance, since the ranking in this case is higher than the simple FRBCSs. Furthermore, when using a good fuzzy rule learning methodology, i.e. the FH-GBML algorithm, the FRBCS approach obtains a better ranking than Ripper and C4.5, even obtaining significant differences in the case of the LTR tuning approach versus the Ripper algorithm.

In the case of data-sets with a high imbalance, regarding FH-GBML we stress that the behaviour of this method is superior to Ripper and C4.5, which is reflected in the ranking value. A more interesting analysis can be carried out in the case of the Chi et al.'s method, where we can clearly observe that, although the ranking is higher in the case of Ripper and C4.5 versus the basic Chi et al.'s method, the use of the 2-tuples genetic tuning enhances significantly the behaviour of the FRBCS; furthermore, in the case of the LTR tuning approach, the fuzzy approach outperforms both algorithms of comparison.

This experimental study supports the conclusion that the 2-tuples based genetic tuning is a solid approach to improve the FRBCS behaviour when dealing with imbalanced data-sets, as it has helped the FRBCS methods to be the best performing algorithms when compared with two classical and well-known algorithms: Ripper and C4.5.

In Fig. 11 we show an example of the use of the 2-tuples genetic tuning with GTS, where the initial and tuned DBs are depicted for the new-thyroid1 data-set. We observe here how the MFs are contextualized for each one of the variables of the problem, adapting the fuzzy system to the problem itself and, in this manner, obtaining better results.

6. Concluding remarks and further work

In this work, we have adapted the 2-tuples based genetic tuning to classification problems with imbalanced data-sets in order to increase the performance of simple FRBCSs.

We have concluded that the tuning step is a necessity, since it always helps FRBCSs to obtain better results. Our empirical and statistical results have shown that the genetic tuning improves the behaviour of the FRBCS in imbalanced data-sets, both globally and for the different types considered, that is, data-sets with a low and high imbalance.

We have also demonstrated that the synergy between the FRBCS and the 2-tuples based genetic tuning is more positive when a good mechanism is chosen to obtain the initial RB.

We must conclude that this approach makes the FRBCSs very competitive in the framework of imbalanced data-sets, outperforming an algorithm of reference in this ambit such as the C4.5 decision tree and Ripper, a classical and accurate rule based algorithm.

Finally, our future work will be oriented to analyse in depth the performance of FRBCSs in the scenario of highly imbalanced data-sets, developing specific learning approaches for dealing with this type of data. Specifically, we are currently studying the generation of the KB by the Genetic Learning of the DB and its potential positive synergy with the genetic 2-tuples tuning.

Acknowledgment

This work had been supported by the Spanish Ministry of Science and Technology under Projects TIN2008-06681-C06-01, TIN2008-06681-C06-02, and the Andalusian Research Plan TIC-3928.

Appendix A. Detailed results for the experimental study

In this appendix we present the complete results tables for all the algorithms used in this work. Thus, the reader can observe the full training and test results, with their associated standard deviation, in order to compare the performance of each approach. In Tables 15 and 16 we show the results for the Chi et al.'s method and for the FH-GBML algorithm with and without 2-tuples based genetic tuning for the original data-sets (without preprocessing). Next, the results with SMOTE preprocessing are shown in Tables 17 and 18, the former for the Chi et al.'s method and the latter for the results of the FH-GBML algorithm, Ripper and C4.5. Finally, Table 19 shows the average number of rules for every single data-set.

Table 16
Complete table of results for FH-GBML with 2-tuples based genetic tuning. Original data-sets.

Data-set	FH-GBML		FH-GBML-GTS		FH-GBML-LTR	
	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
<i>Data-sets with low imbalance</i>						
Glass1	74.22 ± 2.27	70.62 ± 5.64	82.86 ± 3.07	71.29 ± 4.41	85.05 ± 2.89	71.95 ± 9.19
Ecoli0vs1	98.70 ± 0.45	98.00 ± 2.98	98.78 ± 0.41	97.29 ± 2.81	98.94 ± 0.54	96.98 ± 3.40
Wisconsin	97.59 ± 0.34	96.32 ± 1.03	98.59 ± 0.35	96.39 ± 1.55	98.57 ± 0.34	96.01 ± 0.97
Pima	71.38 ± 1.45	69.81 ± 2.01	80.21 ± 2.13	73.48 ± 1.58	81.32 ± 1.32	73.36 ± 2.41
Iris0	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
Glass0	82.30 ± 2.07	80.33 ± 2.61	87.26 ± 1.33	81.02 ± 4.24	87.78 ± 1.74	81.02 ± 4.39
Yeast2	62.75 ± 1.03	61.08 ± 2.65	72.92 ± 1.18	68.54 ± 3.25	74.11 ± 1.89	69.45 ± 2.95
Vehicle2	77.70 ± 2.72	75.99 ± 3.70	95.43 ± 3.31	89.52 ± 3.37	96.59 ± 2.64	90.47 ± 2.80
Vehicle1	64.82 ± 2.26	62.58 ± 2.42	74.80 ± 5.17	65.75 ± 2.53	76.38 ± 6.00	69.11 ± 4.03
Vehicle3	61.20 ± 2.01	58.40 ± 2.68	73.21 ± 4.25	64.64 ± 4.40	76.77 ± 5.88	65.55 ± 6.53
Haberman	60.31 ± 2.38	50.46 ± 2.69	66.30 ± 3.52	49.85 ± 3.98	70.67 ± 6.22	53.28 ± 4.93
Glass0123vs456	94.30 ± 1.89	83.97 ± 6.38	97.85 ± 0.63	90.00 ± 4.88	98.33 ± 0.42	86.62 ± 8.19
Vehicle0	81.94 ± 4.99	75.53 ± 8.03	97.23 ± 1.24	89.70 ± 4.68	97.81 ± 1.08	92.05 ± 2.84
Ecoli1	87.65 ± 3.54	85.22 ± 4.06	93.22 ± 0.70	87.83 ± 4.77	93.76 ± 0.78	91.39 ± 3.35
New-Thyroid2	98.23 ± 1.29	95.75 ± 4.03	99.93 ± 0.16	98.29 ± 3.10	100.0 ± 0.00	96.03 ± 4.20
New-Thyroid1	98.22 ± 1.14	93.73 ± 3.57	100.0 ± 0.00	98.29 ± 3.82	100.0 ± 0.00	97.74 ± 3.57
Ecoli2	90.21 ± 3.71	85.73 ± 5.86	95.28 ± 1.34	87.64 ± 2.77	95.59 ± 1.55	86.74 ± 3.07
Segment0	95.27 ± 1.50	95.65 ± 2.10	99.76 ± 0.32	99.14 ± 0.41	99.63 ± 0.48	99.32 ± 0.42
Glass6	95.19 ± 1.81	87.13 ± 9.51	98.70 ± 1.19	97.46 ± 6.99	99.57 ± 0.97	92.57 ± 4.01
Yeast3	85.27 ± 1.88	84.57 ± 4.19	94.65 ± 0.99	92.49 ± 2.04	94.50 ± 0.89	92.18 ± 1.14
Ecoli3	80.25 ± 8.13	75.48 ± 3.68	92.99 ± 3.35	84.92 ± 10.35	93.47 ± 4.02	86.51 ± 7.38
Page-Blocks0	82.64 ± 2.54	81.64 ± 2.36	91.57 ± 0.80	90.07 ± 0.88	92.13 ± 1.02	90.66 ± 1.67
Mean	83.64 ± 2.25	80.36 ± 3.74	90.52 ± 1.61	84.71 ± 3.49	91.41 ± 1.85	85.41 ± 3.70
<i>Data-sets with high imbalance</i>						
Yeast2vs4	84.90 ± 2.92	81.91 ± 8.78	96.02 ± 2.36	90.15 ± 2.31	95.91 ± 2.21	87.01 ± 3.18
Yeast05679vs4	70.75 ± 5.05	67.35 ± 7.98	81.23 ± 3.96	75.90 ± 11.33	82.06 ± 5.39	70.45 ± 7.33
Vowel0	84.83 ± 5.92	82.05 ± 10.95	94.17 ± 2.27	89.17 ± 6.38	94.11 ± 2.98	89.05 ± 5.94
Glass016vs2	54.40 ± 1.54	48.57 ± 1.75	57.25 ± 4.26	52.19 ± 6.53	57.52 ± 4.11	48.00 ± 1.63
Glass2	54.33 ± 3.06	49.49 ± 0.70	54.40 ± 3.13	50.00 ± 0.00	54.40 ± 3.13	50.00 ± 0.00
Ecoli4	92.97 ± 3.30	89.53 ± 10.61	99.22 ± 1.43	88.42 ± 9.77	98.09 ± 1.75	86.55 ± 8.29
shuttle0vs4	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
yeastB1vs7	64.11 ± 1.73	54.65 ± 6.93	71.52 ± 5.88	55.73 ± 6.54	74.08 ± 5.91	57.63 ± 5.69
Glass4	83.26 ± 4.12	65.67 ± 16.70	96.12 ± 3.79	77.09 ± 21.52	98.09 ± 2.62	87.59 ± 11.86
Page-Blocks13vs4	96.41 ± 3.06	95.43 ± 4.99	99.97 ± 0.06	95.77 ± 5.30	100.0 ± 0.00	97.88 ± 4.12
Abalone9-18	59.15 ± 2.50	53.47 ± 5.06	70.21 ± 3.61	65.61 ± 5.90	70.37 ± 5.78	64.32 ± 3.83
Glass016vs5	83.50 ± 4.95	59.43 ± 13.58	87.86 ± 7.82	59.14 ± 13.84	90.36 ± 6.13	59.14 ± 14.51
shuttle2vs4	100.0 ± 0.00	74.18 ± 23.98	100.0 ± 0.00	74.18 ± 23.98	100.0 ± 0.00	74.18 ± 23.98
Yeast1458vs7	51.65 ± 2.26	49.70 ± 0.67	52.46 ± 3.37	49.77 ± 0.34	52.50 ± 3.42	49.55 ± 0.62
Glass5	71.01 ± 7.54	54.27 ± 11.60	66.79 ± 13.10	49.51 ± 0.67	71.07 ± 17.70	49.02 ± 1.59
Yeast2vs8	77.50 ± 2.61	72.39 ± 13.55	77.50 ± 2.61	72.39 ± 13.55	77.50 ± 2.61	72.39 ± 13.55
Yeast4	54.43 ± 2.62	51.82 ± 4.07	70.63 ± 10.46	65.18 ± 11.10	70.14 ± 11.46	61.62 ± 10.45
Yeast1289vs7	57.08 ± 5.02	51.50 ± 3.82	61.78 ± 6.20	54.45 ± 4.97	62.23 ± 6.56	52.68 ± 4.80
Yeast5	74.63 ± 6.82	72.15 ± 8.28	91.99 ± 13.90	87.88 ± 9.55	92.16 ± 14.00	87.95 ± 10.25
Yeast6	52.49 ± 2.97	51.26 ± 3.10	73.94 ± 21.92	64.71 ± 14.94	70.33 ± 21.00	63.56 ± 12.66
Ecoli0137vs26	85.67 ± 4.94	74.63 ± 24.78	85.67 ± 4.94	74.63 ± 24.78	87.33 ± 5.35	74.63 ± 24.78
Abalone19	50.00 ± 0.00	50.00 ± 0.00	50.00 ± 0.00	50.00 ± 0.00	50.00 ± 0.00	50.00 ± 0.00
Mean	72.87 ± 3.32	65.88 ± 8.27	79.03 ± 5.23	70.09 ± 8.79	79.47 ± 5.55	69.69 ± 7.68
<i>All data-sets</i>						
Global	78.47 ± 2.78	73.35 ± 5.96	84.95 ± 3.39	77.70 ± 6.12	85.58 ± 3.68	77.86 ± 5.57

Table 19

Number of Rules for Chi-3, FH-GBML, Ripper and C4.5 for all data-sets of the study. All models were trained with a balanced training set (preprocessing).

Data-set	Chi-3	FH-GBML	Ripper	C4.5
<i>Data-sets with low imbalance</i>				
Glass1	37.80 ± 0.45	18.80 ± 2.17	12.20 ± 1.10	13.60 ± 2.51
Ecoli0vs1	30.60 ± 1.95	25.80 ± 11.90	3.20 ± 1.30	2.00 ± 0.00
Wisconsin	267.60 ± 3.44	28.60 ± 3.91	9.20 ± 1.30	8.20 ± 0.45
Pima	96.00 ± 3.39	21.20 ± 6.30	25.80 ± 3.03	28.60 ± 4.72
Iris0	14.40 ± 1.14	19.20 ± 0.84	2.20 ± 0.45	2.00 ± 0.00
Glass0	35.60 ± 1.82	16.40 ± 4.28	9.40 ± 2.07	10.20 ± 2.17
Yeast2	93.20 ± 4.02	24.00 ± 11.94	26.80 ± 2.17	44.40 ± 6.80
Vehicle2	382.80 ± 3.56	29.20 ± 4.82	9.40 ± 1.34	23.20 ± 2.59
Vehicle1	349.60 ± 3.36	52.60 ± 34.30	29.60 ± 2.70	59.40 ± 6.91
Vehicle3	340.00 ± 3.00	22.60 ± 5.68	31.40 ± 5.55	63.60 ± 7.99
Haberman	15.20 ± 0.45	17.80 ± 1.30	16.40 ± 1.14	13.80 ± 6.38
Glass0123vs456	44.60 ± 3.51	16.60 ± 2.07	5.20 ± 1.10	7.20 ± 1.30
Vehicle0	351.60 ± 10.36	24.80 ± 1.79	14.20 ± 2.59	28.20 ± 3.19
Ecoli1	47.80 ± 3.35	9.20 ± 2.49	10.80 ± 2.77	6.40 ± 3.58
New-Thyroid2	20.00 ± 1.22	20.40 ± 1.67	3.80 ± 0.84	6.80 ± 0.84
New-Thyroid1	20.00 ± 1.00	19.40 ± 2.61	4.60 ± 0.55	6.00 ± 1.87
Ecoli2	48.60 ± 1.34	13.00 ± 2.55	10.20 ± 2.86	17.40 ± 2.70
Segment0	294.60 ± 4.10	16.20 ± 1.48	7.00 ± 0.71	12.80 ± 2.68
Glass6	46.80 ± 1.92	19.00 ± 3.74	5.20 ± 0.84	9.00 ± 1.87
Yeast3	99.20 ± 4.38	10.80 ± 3.56	26.20 ± 3.96	36.60 ± 4.22
Ecoli3	48.40 ± 1.34	12.20 ± 2.05	8.40 ± 2.79	14.20 ± 3.35
Page-Blocks0	59.00 ± 3.00	18.80 ± 2.77	59.40 ± 2.30	110.60 ± 4.22
Mean	124.70 ± 2.82	20.75 ± 5.19	15.03 ± 1.98	23.83 ± 3.20
<i>Data-sets with high imbalance</i>				
Yeast2vs4	43.00 ± 2.55	16.80 ± 10.99	16.00 ± 4.00	20.40 ± 3.13
Yeast05679vs4	63.40 ± 5.18	13.80 ± 3.42	22.00 ± 4.00	30.20 ± 2.95
Vowel0	323.20 ± 8.56	30.40 ± 22.30	7.20 ± 1.10	15.80 ± 3.49
Glass016vs2	32.60 ± 1.52	17.20 ± 1.79	11.00 ± 1.58	15.80 ± 3.96
Glass2	33.20 ± 3.27	16.60 ± 1.67	10.00 ± 1.22	15.20 ± 4.66
Ecoli4	46.80 ± 2.59	10.80 ± 1.79	5.40 ± 1.52	8.00 ± 2.92
Shuttle0vs4	25.80 ± 4.66	49.80 ± 0.45	2.80 ± 0.45	2.00 ± 0.00
yeast1vs7	70.80 ± 5.40	19.80 ± 6.98	23.20 ± 3.83	32.20 ± 8.35
Glass4	42.20 ± 6.30	23.60 ± 9.76	4.20 ± 1.30	10.40 ± 2.07
Page-Blocks13vs4	64.80 ± 5.54	16.80 ± 3.56	6.00 ± 1.73	6.20 ± 1.79
Abalone9-18	43.40 ± 2.41	17.20 ± 11.10	25.20 ± 1.64	63.00 ± 11.45
Glass016vs5	48.00 ± 4.74	17.60 ± 3.51	6.00 ± 1.00	8.60 ± 1.52
shuttle2vs4	11.00 ± 2.00	17.40 ± 18.32	4.40 ± 0.89	4.40 ± 0.89
Yeast1458vs7	80.20 ± 5.72	14.20 ± 3.11	27.60 ± 2.51	48.80 ± 7.16
Glass5	41.60 ± 3.29	15.00 ± 3.39	5.00 ± 0.71	9.40 ± 1.14
Yeast2vs8	40.80 ± 2.59	14.20 ± 5.97	11.40 ± 1.67	28.00 ± 5.15
Yeast4	86.20 ± 3.35	21.60 ± 11.33	25.80 ± 3.27	58.60 ± 5.27
Yeast1289vs7	78.00 ± 4.64	16.40 ± 2.51	29.80 ± 1.92	57.60 ± 6.66
Yeast5	101.40 ± 4.04	12.40 ± 3.29	9.40 ± 1.52	18.20 ± 0.84
Yeast6	87.40 ± 3.65	12.20 ± 1.92	13.20 ± 1.30	40.20 ± 6.46
Ecoli0137vs26	77.80 ± 5.07	15.80 ± 6.30	7.40 ± 1.95	7.60 ± 1.52
Abalone19	69.20 ± 3.49	18.60 ± 3.71	30.20 ± 1.30	121.20 ± 9.12
Mean	68.67 ± 4.12	18.55 ± 6.24	13.78 ± 1.84	28.26 ± 4.11
<i>All data-sets</i>				
Global	96.69 ± 3.47	19.65 ± 5.71	14.40 ± 1.91	26.05 ± 3.66

References

- [1] R. Alcalá, J. Alcalá-Fdez, J. Casillas, O. Cordón, F. Herrera, Hybrid learning models to get the interpretability-accuracy trade-off in fuzzy modeling, *Soft Computing* 10 (9) (2006) 717–734.
- [2] R. Alcalá, J. Alcalá-Fdez, F. Herrera, A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection, *IEEE Transactions on Fuzzy Systems* 15 (4) (2007) 616–635.
- [3] R. Alcalá, J. Alcalá-Fdez, F. Herrera, J. Otero, Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation, *International Journal of Approximate Reasoning* 44 (2007) 45–64.
- [4] R. Alcalá, J.M. Benítez, J. Casillas, O. Cordón, R. Pérez, Fuzzy control of HVAC systems optimized by genetic algorithms, *Applied Intelligence* 18 (2003) 155–177.
- [5] J. Alcalá-Fdez, F. Herrera, F.A. Márquez, A. Peregrín, Increasing fuzzy rules cooperation based on evolutionary adaptive inference systems, *International Journal of Intelligent Systems* 22 (9) (2007) 1035–1064.
- [6] J. Alcalá-Fdez, L. Sánchez, S. García, M. del Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernández, F. Herrera, KEEL: a software tool to assess evolutionary algorithms to data mining problems, *Soft Computing* 13 (3) (2009) 307–318.
- [7] A. Asuncion, D. Newman, UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences, 2007. URL: <<http://www.ics.uci.edu/mllearn/MLRepository.html>>.

- [8] R. Barandela, J.S. Sánchez, V. García, E. Rangel, Strategies for learning in class imbalance problems, *Pattern Recognition* 36 (3) (2003) 849–851.
- [9] A. Bastian, How to handle the flexibility of linguistic variables with applications, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 2 (4) (1994) 463–484.
- [10] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A study of the behaviour of several methods for balancing machine learning training data, *SIGKDD Explorations* 6 (1) (2004) 20–29.
- [11] A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, *Pattern Recognition* 30 (7) (1997) 1145–1159.
- [12] J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.), *Accuracy Improvements in Linguistic Fuzzy Modeling*, Springer-Verlag, 2003.
- [13] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, *Journal of Artificial Intelligent Research* 16 (2002) 321–357.
- [14] N.V. Chawla, N. Japkowicz, A. Kolcz, Editorial: special issue on learning from imbalanced data sets, *SIGKDD Explorations* 6 (1) (2004) 1–6.
- [15] M.-C. Chen, L.-S. Chen, C.-C. Hsu, W.-R. Zeng, An information granulation based data mining approach for classifying imbalanced data, *Information Sciences* 178 (16) (2008) 3214–3227.
- [16] Z. Chi, H. Yan, T. Pham, *Fuzzy Algorithms with Applications to Image Processing and Pattern Recognition*, World Scientific, 1996.
- [17] W.W. Cohen, Fast effective rule induction, in: *Machine Learning: Proceedings of the Twelfth International Conference*, Morgan Kaufmann, 1995, pp. 115–123.
- [18] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends, *Fuzzy Sets and Systems* 141 (1) (2004) 5–31.
- [19] O. Cordón, F. Herrera, A three-stage evolutionary process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples, *International Journal of Approximate Reasoning* 17 (4) (1997) 369–407.
- [20] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, *Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, *Advances in Fuzzy Systems – Applications and Theory*, vol. 19, World Scientific, 2001.
- [21] O. Cordón, F. Herrera, P. Villar, A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base, *Information Sciences* 136 (1–4) (2001) 85–107.
- [22] K.A. Crockett, Z. Bandar, J. Fowdar, J. O’Shea, Genetic tuning of fuzzy inference within fuzzy classifier systems, *Expert Systems* 23 (2) (2006) 63–82.
- [23] K.A. Crockett, Z. Bandar, D. Mclean, J. O’Shea, On constructing a fuzzy inference framework using crisp decision trees, *Fuzzy Sets and Systems* 157 (2006) 2809–2832.
- [24] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research* 7 (2006) 1–30.
- [25] P. Domingos, Metacost: a general method for making classifiers cost sensitive, in: *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 155–164.
- [26] D. Driankov, H. Hellendoorn, M. Reinfrank, *An Introduction to Fuzzy Control*, Springer, 1993.
- [27] C. Drummond, R.C. Holte, C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling, in: *Proceedings of the International Conference on Machine Learning (ICML 2003) Workshop on Learning from Imbalanced Data Sets II*, 2003.
- [28] P. Ducange, B. Lazzarini, F. Marcelloni, Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets, *Soft Computing*, in press. doi: 10.1007/s00500-009-0460-y.
- [29] L.J. Eshelman, *Foundations of Genetic Algorithms*, Ch. The CHC Adaptive Search Algorithm: How to have Safe Search When Engaging in Nontraditional Genetic Recombination, Morgan Kaufman, 1991, pp. 265–283.
- [30] A. Estabrooks, T. Jo, N. Japkowicz, A multiple resampling method for learning from imbalanced data sets, *Computational Intelligence* 20 (1) (2004) 18–36.
- [31] A. Fernández, M.J. del Jesus, F. Herrera, Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets, *International Journal of Approximate Reasoning* 50 (2009) 561–577.
- [32] A. Fernández, M.J. del Jesus, F. Herrera, On the influence of an adaptive inference system in fuzzy rule based classification systems for imbalanced data-sets, *Expert Systems with Applications* 36 (6) (2009) 9805–9812.
- [33] A. Fernández, S. García, M.J. del Jesus, F. Herrera, A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets, *Fuzzy Sets and Systems* 159 (18) (2008) 2378–2398.
- [34] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, *Soft Computing* 13 (10) (2009) 959–977.
- [35] S. García, F. Herrera, An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons, *Journal of Machine Learning Research* 9 (2008) 2677–2694.
- [36] V. García, R. Mollineda, J.S. Sánchez, On the k-NN performance in a challenging scenario of imbalance and overlapping, *Pattern Analysis Applications* 11 (3–4) (2008) 269–280.
- [37] F. Herrera, Genetic fuzzy systems: taxonomy, current research trends and prospects, *Evolutionary Intelligence* 1 (2008) 27–46.
- [38] F. Herrera, M. Lozano, A.M. Sánchez, A taxonomy for the crossover operator for real-coded genetic algorithms: an experimental study, *International Journal of Intelligent Systems* 18 (2003) 309–338.
- [39] F. Herrera, M. Lozano, J.L. Verdegay, Tuning fuzzy logic controllers by genetic algorithms, *International Journal of Approximate Reasoning* 12 (1995) 299–315.
- [40] F. Herrera, L. Martínez, A 2-tuple fuzzy linguistic representation model for computing with words, *IEEE Transactions on Fuzzy Systems* 8 (6) (2000) 746–752.
- [41] J. Huang, C.X. Ling, Using AUC and accuracy in evaluating learning algorithms, *IEEE Transactions on Knowledge and Data Engineering* 17 (3) (2005) 299–310.
- [42] Y.M. Huang, C.M. Hung, H.C. Jiau, Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem, *Nonlinear Analysis: Real World Applications* 7 (4) (2006) 720–747.
- [43] H. Ishibuchi, T. Nakashima, M. Nii, *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*, Springer-Verlag, 2004.
- [44] H. Ishibuchi, T. Yamamoto, Rule weight specification in fuzzy rule-based classification systems, *IEEE Transactions on Fuzzy Systems* 13 (2005) 428–435.
- [45] H. Ishibuchi, T. Yamamoto, T. Nakashima, Hybridization of fuzzy GBML approaches for pattern classification problems, *IEEE Transactions on System, Man and Cybernetics B* 35 (2) (2005) 359–365.
- [46] J. Jang, ANFIS: adaptive network based fuzzy inference system, *IEEE Transactions on System, Man and Cybernetics* 23 (3) (1993) 665–684.
- [47] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, *Intelligent Data Analysis* 6 (5) (2002) 429–450.
- [48] C. Karr, Genetic algorithms for fuzzy controllers, *AI Expert* 6 (2) (1991) 26–33.
- [49] K. Kilić, O. Uncu, I.B. Türksen, Comparison of different strategies of utilizing fuzzy clustering in structure identification, *Information Sciences* 177 (23) (2007) 5153–5162.
- [50] Z. Lei, L. Ren-hou, Designing of classifiers based on immune principles and fuzzy rules, *Information Sciences* 178 (7) (2008).
- [51] M. Li, Z. Wang, A hybrid coevolutionary algorithm for designing fuzzy classifiers, *Information Sciences* 179 (12) (2009) 1970–1983.
- [52] Y.-H. Liu, Y.-T. Chen, Face recognition using total margin-based adaptive fuzzy support vector machines, *IEEE Transactions on Neural Networks* 18 (1) (2007) 178–192.
- [53] F.A. Márquez, A. Peregrín, F. Herrera, Cooperative evolutionary learning of fuzzy rules and parametric aggregation connectors for mamdani linguistic fuzzy systems, *IEEE Transactions on Fuzzy Systems* 15 (6) (2007) 1162–1178.

- [54] M. Mazurowski, P. Habas, J. Zurada, J. Lo, J. Baker, G. Tourassi, Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance, *Neural Networks* 21 (2–3) (2008) 427–436.
- [55] A. Orriols-Puig, E. Bernadó-Mansilla, Evolutionary rule-based systems for imbalanced datasets, *Soft Computing* 13 (3) (2009) 213–225.
- [56] X. Peng, I. King, Robust BPPM training based on second-order cone programming and its application in medical diagnosis, *Neural Networks* 21 (2–3) (2008) 450–457.
- [57] C. Phua, D. Alahakoon, V. Lee, Minority report in fraud detection: classification of skewed data, *SIGKDD Explorations Newsletter* 6 (1) (2004) 50–59.
- [58] F. Provost, T. Fawcett, Robust classification for imprecise environments, *Machine Learning* 42 (3) (2001) 203–231.
- [59] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [60] D. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman & Hall/CRC, 2003.
- [61] C.-T. Su, L.-S. Chen, Y. Yih, Knowledge acquisition through information granulation for imbalanced data, *Expert Systems with Applications* 31 (2006) 531–541.
- [62] C.-T. Su, Y.-H. Hsiao, An evaluation of the robustness of MTS for imbalanced data, *IEEE Transactions on Knowledge Data Engineering* 19 (10) (2007) 1321–1332.
- [63] Y. Sun, M.S. Kamel, A.K. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, *Pattern Recognition* 40 (2007) 3358–3378.
- [64] S. Suresh, N. Sundararajan, P. Saratchandran, Risk-sensitive loss functions for sparse multi-category classification problems, *Information Sciences* 178 (12) (2008) 2621–2638.
- [65] L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, *IEEE Transactions on Systems, Man, and Cybernetics* 25 (2) (1992) 353–361.
- [66] G. Weiss, F. Provost, Learning when training data are costly: the effect of class distribution on tree induction, *Journal of Artificial Intelligence Research* 19 (2003) 315–354.
- [67] G.M. Weiss, Mining with rarity: a unifying framework, *SIGKDD Explorations* 6 (1) (2004) 7–19.
- [68] F. Wilcoxon, Individual comparisons by ranking methods, *Biometrics* 1 (1945) 80–83.
- [69] D.L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, *IEEE Transactions on Systems, Man, and Communications* 2 (3) (1972) 408–421.
- [70] G. Wu, E. Chang, KBA: kernel boundary alignment considering imbalanced data distribution, *IEEE Transactions on Knowledge Data Engineering* 17 (6) (2005) 786–795.
- [71] L. Xu, M.Y. Chow, L.S. Taylor, Power distribution fault cause identification with imbalanced data using the data mining-based fuzzy classification E-Algorithm, *IEEE Transactions on Power Systems* 22 (1) (2007) 164–171.
- [72] Q. Yang, X. Wu, 10 challenging problems in data mining research, *International Journal of Information Technology and Decision Making* 5 (4) (2006) 597–604.
- [73] Z.-H. Zhou, X.-Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, *IEEE Transactions on Knowledge Data Engineering* 18 (1) (2006) 63–77.
- [74] M.J. Zolghadri, E.G. Mansoori, Weighting fuzzy classification rules using receiver operating characteristics (ROC) analysis, *Information Sciences* 177 (11) (2007) 2296–2307.