International Journal of Uncertainty,
Fuzziness and Knowledge-Based Systems
Vol. 20, No. 3 (2012) 369–397

DOI: 10.1142/S0218488512500195

World Scientific
www.worldscientific.com

FEATURE SELECTION AND GRANULARITY LEARNING IN GENETIC FUZZY RULE-BASED CLASSIFICATION SYSTEMS FOR HIGHLY IMBALANCED DATA-SETS

PEDRO VILLAR

Department of Software Engineering, University of Granada, ETSIIT, 18071 Granada, Spain
pvillarc@ugr.es

ALBERTO FERNÁNDEZ

Department of Computer Science, University of Jaén, 23071 Jaén, Spain
alberto.fernandez@ujaen.es

RAMÓN A. CARRASCO

Department of Software Engineering, University of Granada, ETSIIT, 18071 Granada, Spain
racg@ugr.es

FRANCISCO HERRERA

Department of Computer Science and Artificial Intelligence, University of Granada, ETSIIT, 18071 Granada, Spain
herrera@decsai.ugr.es

Received 17 August 2010
Revised 30 March 2012

This paper proposes a Genetic Algorithm for jointly performing a feature selection and granularity learning for Fuzzy Rule-Based Classification Systems in the scenario of highly imbalanced data-sets. We refer to imbalanced data-sets when the class distribution is not uniform, a situation that it is present in many real application areas. The aim of this work is to get more compact models by selecting the adequate variables and adapting the number of fuzzy labels for each problem, improving the interpretability of the model. The experimental analysis is carried out over a wide range of highly imbalanced data-sets and uses the statistical tests suggested in the specialized literature.

Keywords: Fuzzy rule-based classification systems; imbalanced data-sets; genetic algorithms; feature selection; granularity level.

1. Introduction

The problem of imbalanced data-sets[1–3] for binary classification occurs when the number of instances for each class are very different between them, which can lead to a good classification of the majority class and a poor accuracy on the minority examples. Furthermore, the less representative class is usually the one which has more interest from the point of view of the learning task.[4–6] We must stress the importance of imbalanced data-sets, since such type of data appears in most of the real domains of classification. Some examples are face recognition,[7] remote-sensing[8] and forecasting of ozone levels[9,10] among others.

We develop an experimental analysis in the context of imbalance classification. In this study, we will make use of linguistic Fuzzy Rule Based Classification Systems (FRBCSs), a very useful tool in the framework of computational intelligence, since they provide a very interpretable model for the end user.[11] The good behavior of FRBCS when dealing with imbalanced data-sets has been recently analyzed in Ref. 12.

An FRBCS presents two main components: the Inference System and the Knowledge Base (KB). The KB is composed of the Rule Base (RB) constituted by the collection of fuzzy rules, and of the Data Base (DB), that comprises the number of labels for each linguistic variable (granularity level), as well as the fuzzy membership functions associated to each label (fuzzy partition). The composition of the KB of an FRBCS directly depends on the problem being solved. If there is no expert information about the problem under solving, an automatic learning process must be used to derive the KB from examples.

The number of labels per linguistic variable (granularity) is an information that has not been considered to be relevant for the majority of FRBCS learning methods. The usual way to proceed involves choosing a number of linguistic terms for each linguistic variable, which is normally the same for all of them (the most used values are the odd numbers between 3 and 7). This operation mode makes the granularity level has a significant influence on the FRBCS performance. The fuzzy partition granularity of a linguistic variable can be viewed as a sort of context information with a significative influence in the FRBCS behavior. Considering a specific label set for a variable, some labels can result irrelevant, that is, they can contribute nothing and even can cause confusion. In other cases, it would be necessary to add new labels to appropriately differentiate the values of the variable. The high influence of granularity in fuzzy modeling has been analyzed in Ref. 13 and some approaches for automatic learning of the KB in fuzzy modeling and fuzzy classification include the granularity learning.[14–17] In a previous work,[18] we analyze the influence of granularity learning in the performance of FRBCSs for imbalanced data-sets, and the results obtained show that a significant improvement in the classification ability is possible just by learning an adequate number of labels per variable although the complexity of the model was lightly increased.

On the other hand, in many classification problems, a large number of features can originate RBs with a high number of rules, thus presenting a low degree of interpretability and a possible overfitting (the error over the training data-set is very low but the FRBCS present a significative decrease on the prediction ability). This problem can be addressed from a double perspective:

- Via the compactness and reduction of the rule set, minimizing the number of fuzzy rules included in it.
- Via a feature selection process that reduces the number of features used by the FRBCS.

Rule reduction methods have been formulated using different approaches: Neural Networks,[19] clustering techniques,[20] orthogonal transformation methods,[21] similarity measures[22,23] and Genetic Algorithms (GAs).[24,25] Notice that, for high dimensional problems and problems where a high number of instances is available, it is difficult for rule reduction approaches to get small rule sets, and therefore the system comprehensibility and interpretability may not be as good as desired. For high dimensionality classification problems, a feature selection process, that determines the most relevant variables before or during the FRBCS inductive learning process, must be considered.[26] It increases the efficiency and accuracy of the learning and classification stages.

The main objective of this paper is to propose a genetic learning process to derive the KB of a FRBCS for imbalanced data-sets in order to maintain the improvement level of the prediction ability achieved in Ref. 18 (by the granularity learning) joint with a significative reduction of the model complexity in order to increase the FRBCS interpretability (by the feature selection).

Our proposal uses a GA for jointly perform a feature selection and a granularity learning, and considers a classical FRBCS learning method to derive the RB, the Chi *et al.*'s approach.[27] This KB generation approach, in which A DB generation process wraps a RB learning one, is composed of two different (and independent) learning processes. Therefore, our proposal of GA for DB generation can be combined with any RB generation method. We have chosen the Chi *et al.*'s method for its simplicity but more accurate ones can be used.

In order to show the influence of choosing a good set of features and an adequate granularity level, we compare the results obtained with the ones obtained by Chi *et al.*'s method with all the variables selected with and without a granularity learning process. We also want to check the performance of our proposal compared with a non-FRBCS classification model, C4.5,[28] a decision tree algorithm that has been used as a reference in the imbalanced data-sets field.[29–32]

We have selected a large collection of imbalanced data-sets from KEEL data-set repository[1][33] for developing our experimental analysis. In order to deal with the problem of imbalanced data-sets we will make use of a preprocessing

[1]`http://www.keel.es/dataset.php`

technique, the "Synthetic Minority Over-sampling Technique" (SMOTE),[34] to balance the distribution of training examples in both classes. Furthermore, we will perform a statistical study using non-parametric tests[35–37] to find significant differences among the obtained results.

This paper is organized as follows. First, Sec. 2 introduces the preliminary concepts of FRBCSs and imbalanced data-sets used in this paper. Next, in Sec. 3 we will expose the main characteristics of our proposal, a GA for feature selection and granularity learning in FRBCS. The next section describes the experimental study. Finally, in Sec. 5, some conclusions will be pointed out.

2. Preliminaries

This section first introduces some basic concepts about FRBCS and describes the fuzzy rule learning algorithm used in our work. Next, the problem of imbalanced data-sets is addressed in detail.

2.1. *Fuzzy rule based classification systems*

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

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

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

where R_j is the label of the jth rule, $x = (x_1, \ldots, x_n)$ 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.[38] We use triangular MFs as antecedent fuzzy sets.

In order to build the RB, we have chosen a classical and simple FRBCS, following the same scheme as our previous works:[12,39,40] the Chi *et al.*'s rule generation method.[27] This FRBCSs design method is an extension of the well-known Wang and Mendel method[41] to classification problems. To generate the fuzzy RB, this method 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* $x_p = (x_{p1}, \ldots, x_{pn}, C_p)$. 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.

2.2. *Basic concepts on imbalanced data-sets*

In this section, we first introduce the problem of imbalanced data-sets. Then, we describe the pre-processing technique we have applied in order to deal with the imbalanced data-sets: the SMOTE algorithm.[34] Finally, we will present the evaluation metrics for this type of classification problem.

2.2.1. *The problem of imbalanced data-sets*

The main property of this type of classification problem (in a binary context) is that the examples of one class outnumbers examples of the other one.[1,3] Since most of the standard learning algorithms consider a balanced training set, this situation may cause the obtention of suboptimal classification models, i.e. a good coverage of the majority examples whereas the minority ones are misclassified more frequently.[2,3]

The reasons behind this behaviour include:

- The use of the standard accuracy metric, which is independent of the class distribution and therefore it favours the coverage of the majority class examples.
- The small disjuncts that can be found in the data set[42] and the difficulty of most learning algorithms in detecting these areas.[30,43] In fact, learning algorithms try to benefit those models with a higher degree of coverage and these small disjuncts imply the application of very specific models which are discarded in favor or more general ones.
- Related to the apparition of small disjuncts, we must stress the overlapping between the examples of the positive and the negative class,[44] in which the minority class examples can be simply treated as noise and ignored by the learning algorithm. These phenomena are depicted in Fig. 1(a) and 1(b) respectively.

In the specialized literature, researchers usually manage all imbalanced data sets as a whole.[29,45,46] Nevertheless, in this paper we organize the different data sets according to their degree of imbalance using the imbalance ratio (IR),[30] which is defined as the ratio of the number of instances of the majority class and the minority

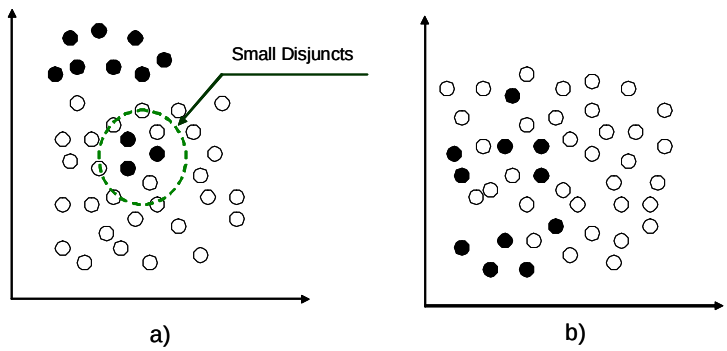


Fig. 1. Example of the imbalance between classes: (a) small disjuncts (b) overlapping between classes.

class. Therefore we can focus only in those problem with a high IR considering them more interesting from a learning point of view.

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[45,47–49] and external approaches that preprocess the data in order to diminish the effect of their class imbalance.[29,50] 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.[51–53]

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,[12] we analysed the cooperation of some preprocessing methods with FRBCSs, showing a good behaviour for the oversampling methods, especially in the case of the SMOTE methodology.[34] 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.

2.2.2. *Pre-processing imbalanced data sets. The SMOTE algorithm*

In the specialized literature, we can find several papers about resampling techniques studying the effect of changing class distribution to deal with imbalanced data-sets. Those works have proved empirically that, applying a preprocessing step in order to balance the class distribution, is usually a positive solution.[12,29,54] Furthermore, the main advantage of these techniques is that they are independent of the underlying classifier.

Resampling techniques can be categorized into three groups or families:

(1) *Undersampling methods*, which create a subset of the original data-set by eliminating instances (usually majority class instances).
(2) *Oversampling methods*, which create a superset of the original data-set by replicating some instances or creating new instances from existing ones.
(3) *Hybrids methods*, which combine both sampling approaches.

As mentioned before, previous analysis on preprocessing methods with FRBCSs have shown the goodness of the oversampling techniques. The simplest approach, random oversampling, makes exact copies of existing instances, and therefore several authors agree that this method can increase the likelihood of occurring overfitting.[29] According to the previous fact, more sophisticated methods have been proposed based on the generation of synthetic samples. Among them, the SMOTE methodology,[34] whose main idea is to form new minority class examples by interpolating between several minority class examples that lie together, have become one of the most significant approaches in this area.

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 randomised interpolation. The implementation of this work uses only one nearest neighbour with the Euclidean distance, and balances both classes to 50% distribution.

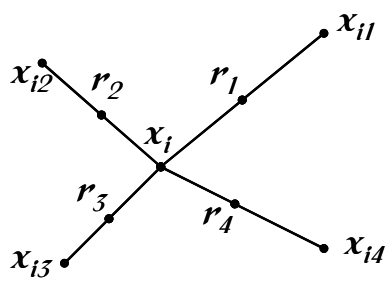


Fig. 2. An illustration of how to create the synthetic data points in the SMOTE algorithm.

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. A numerical example is detailed in Fig. 3.

```
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) + rand(0-1) * (-2,-1) rand(0-1)
generates a random number between 0 and 1.
```

Fig. 3. Example of the SMOTE application.

2.2.3. *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.

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)

The most used empirical measure, accuracy (2), cannot be considered for imbalanced data sets, since it does not distinguish between the number of correct classifications of the different classes, which may lead to erroneous conclusions in this case. As a classical example, if the ratio of imbalance presented in the data is 1:100, i.e. there is one positive instance versus ninety-nine negatives, a classifier that obtains an accuracy rate of 99% is not truly accurate if it does not correctly cover any minority class instance.

$$Acc = \frac{TP + TN}{TP + FN + FP + TN}\,. \tag{2}$$

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

Since in this classification scenario we intend to achieve good quality results for both classes, there is a necessity of obtaining one way to combine the individual measures of both the positive and negative classes, being none of these measures alone adequate by itself.

Specifically, a well-known approach to unify these measures and to produce an evaluation criteria is to use the Receiver Operating Characteristic (ROC) graphic.[55] This graphic allows to visualize the trade-off between the benefits (TP_{rate}) and costs (FP_{rate}), thus it evidences that any classifier cannot increase the number of true positives without also increasing the false positives. The Area Under the ROC Curve (AUC)[56] corresponds to the probability of correctly identifying which one of the two stimuli is noise and which one is signal plus noise. AUC provides a single measure of a classifier's performance for evaluating which model is better on average.

Figure 4 shows how to build the ROC space plotting on a two-dimensional chart the TP_{rate} (Y-axis) against the FP_{rate} (X-axis). Points in $(0, 0)$ and $(1, 1)$ are trivial classifiers where the predicted class is always the negative and positive respectively. On the contrary, $(0, 1)$ point represents the perfect classification. AUC measure is computed just by obtaining the area of the graphic as:

$$\text{AUC} = \frac{1 + TP_{\text{rate}} - FP_{\text{rate}}}{2} \,. \tag{3}$$

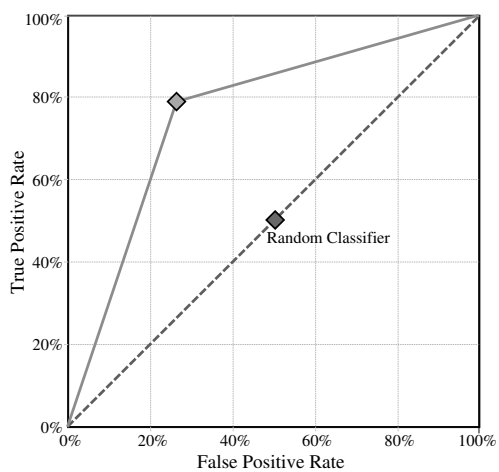


Fig. 4. Example of an ROC plot. Two classifiers' curves are depicted: the dashed line represents a random classifier, whereas the solid line is a classifier which is better than the random classifier.

As a final remark, we must state that AUC has a special statistical meaning: it represents the probability that a randomly chosen negative example will have a smaller estimated probability of belonging to the positive class than a randomly chosen positive example.[57] Moreover, AUC also equals to the quantity of Wilcoxon statistic.[58] Please refer to[56] for details.

3. Genetic Algorithm for Feature Selection and Granularity Learning

In this section, we propose an standard generational GA for the DB definition that allows us to select a set of variables (feature selection) and learn an adequate number of labels for each selected variable (granularity learning). We denote our proposal as GA-FS+GL (Genetic Algorithm for Feature Selection and Granularity Learning). In this contribution, the possible values considered for the granularity are taken from the set $\{2, \ldots, 7\}$. Once the granularity for each feature is determined, the DB is built. Uniform partitions with triangular membership functions are considered due to its simplicity. Next, we use a quick method that derives the fuzzy classification rules and then the whole KB is obtained. We must recall from the previous section that the RB learning algorithm used in this work is the method proposed in Ref. 27, that we have called the Chi *et al.*'s rule generation method.

The main purpose of GA-FS+GL is to obtain FRBCSs with good accuracy and reduced complexity taking the feature selection and granularity learning as a base. Unfortunately, FRBCSs with good performance have a high number of rules, thus presenting a low degree of readability. On the other hand, as mentioned before, the KB design methods sometimes lead to a certain overfitting to the training data-set used for the learning process. In order to avoid that problem, our genetic process try to design a compact and interpretable KB by penalizing FRBCSs with high number of selected variables and/or high granularity average as it will be explained in Section 3.3. Next, we describe the components of GA-FS+GL.

3.1. *Encoding the DB*

For a classification problem with N features, each chromosome will be composed of two parts to encode the relevant variables and the number of linguistic terms for variable (i.e. the granularity):

- Relevant variables (C_V): the selected features are stored in a binary coded array of length N. In this array, an 1 indicates that the correspondent variable is selected for the FRBCS.
- Granularity level (C_G): the number of labels per variable is stored in an integer array of length N. The possible values are taken from the set $\{2, \ldots, 7\}$.

If v_i is the bit that represents whether the variable i is selected and g_i is the granularity of variable i, a representation of the chromosome is shown next:

$$C_V = (v_1, v_2, \ldots, v_N) \quad C_G = (g_1, g_2, \ldots, g_N)$$

$$C = (v_1, v_2, \ldots, v_N, g_1, g_2, \ldots, g_N)$$

It would be possible to merge both parts considering only an integer array and including the value 1 as a placeholder for not using the variable. We propose the former coding scheme to assign the same importance to both parts and to make easy the possibility of removing features.

3.2. *Initial gene pool*

The initial population is composed of six groups with a different number of selected variables. Let g the cardinality of the significant term set for the C_V part, in our case $g = 6$, corresponding to the six possibilities for the number of labels, $2 \ldots 7$. The generation of the initial population is described below:

- In the first group all the chromosomes have all the features selected that is, $C_V = (1, 1, 1, \ldots, 1)$. It is composed of $g + 10$ chromosomes (16 in our case). The first g individuals have the same granularity in all its variables. For each granularity level, one individual is created. In the second 10 chromosomes the granularity level is randomly selected.
- The next four groups have the same structure as the first group but each one of them with a different percentage of randomly selected variables (75%, 50%, 25% and 10%). So, each group has $g + 10$ chromosomes (16 in our case).
- The last group is composed for the remaining chromosomes, and all of their components are randomly selected.

The minimum number of individuals is the sum of the chromosomes of the five first groups: $(g + 10) \times 5$ (80 for our proposal). In our case, the total population length is 100. Therefore, the last group is comprised by 20 chromosomes. We try to cover a wide zone of the search space with this initial population.

3.3. *Evaluating the chromosome*

There are three steps that must be done to evaluate each chromosome:

- Generate the DB using the information contained in the chromosome. For all the selected variables ($v_i = 1$), a uniform fuzzy partition with triangular membership functions is built considering the number of labels of that variable (g_i).
- Generate the RB by running the the Chi *et al.*'s method.
- Calculate the value of the evaluation function: We will employ a fitness function composed of the aggregated sum of two values (one of them is an accuracy measure an the other one is a complexity measure), similar to other fitness functions proposed for Genetic Fuzzy Systems in the specialized literature.[14,15,24] The goal of this type of function is to avoid the possible overfitting and to promote the removal of unnecessary features. In our case, we will lightly penalize FRBCSs with high number of selected variables and/or high granularity levels. The fitness function to be minimized is:

$$F_C = \omega_1 \cdot (1 - \text{AUC}) + \omega_2 \cdot (Ng/N)$$

being Ng the sum of the granularity levels of all the selected variables. In order to normalize these two values, we calculate ω_2 taking two values as a base: the AUC of the FRBCS obtained with the RB generation method considering the DB with all the variables selected, the maximum number of labels ($\max\_g$) per

variable and uniform fuzzy partitions:

$$\omega_2 = \alpha_{\omega_2} \cdot \frac{\mathrm{AUC}_{\max\_g}}{\max\_g}$$

with α_{ω_2} being a weighting percentage ($\alpha_{\omega_2} = 1 - \omega_1$). In our case $\max\_g = 7$.

Let see an example of the fitness value calculation. Suppose we have a problem with eight features and we choose $\omega_1 = 0.6$ (consequently, $\alpha_{\omega_2} = 0.4$). Suppose we have run the RB generation method with all the variables selected and the maximum number of labels (seven) in each fuzzy partition and we have calculated the AUC (AUC = 0.8). Then, the value of ω_2 can be calculated:

$$\omega_2 = 0.4 \cdot \frac{0.8}{7} = 0.04571$$

The former calculus is only performed once, before starting the GA. Suppose we have the following chromosome:

$$C = (1, 0, 0, 1, 1, 1, 0, 0, 3, 2, 5, 6, 2, 7, 4, 2)$$

The features selected are 1,4,5 and 6 , each one of them with their correspondent number of labels (3,6,2,7). So, we can calculate the Ng value ($Ng = 3{+}6{+}2{+}7 = 18$). We run the RB generation method with the former selected features and labels, obtaining AUC = 0.74. Then, the fitness value of that chromosome is:

$$F_C = 0.6 \cdot (1 - 0.74) + 0.04571 \cdot (18/8) = 0.156 + 0.103 = 0.259$$

We must note that using the Ng value allows the following situations are considered as equivalent:

— Selection of 6 features of granularity 2
— Selection of 4 features of granularity 3
— Selection of 3 features of granularity 4
— Selection of 2 features of granularity 6

3.4. *Genetic operators*

The following operators are considered.

3.4.1. *Selection*

We will employ the tournament selection with $k = 2$, in which two chromosomes are selected at random from the population, and the one with highest fitness is taken to be included in the next population, after the application of the genetic operators.

3.4.2. *Crossover*

The crossover works in the two parts of the chromosome at the same time. Therefore, an standard crossover operator is applied over C_V and C_G. This operator performs

as follows: a crossover point p is randomly generated and the two parents are crossed at the p-th variable (the possible values for p are $\{2, \ldots, N\}$). The crossover is developed this way in the two chromosome parts, C_V and C_G, thereby producing two meaningful descendants. Let us look at an example in order to clarify the standard crossover application. Let

$$C_1 = (v_1, \ldots, v_p, v_{p+1}, \ldots, v_N, g_1, \ldots, g_p, g_{p+1}, \ldots, g_N)$$
$$C_2 = (v_1^{'}, \ldots, v_p^{'}, v_{p+1}^{'}, \ldots, v_N^{'}, g_1^{'}, \ldots, g_p^{'}, g_{p+1}^{'}, \ldots, g_N^{'})$$

be the individuals to be crossed at point p, the two resulting offspring are:

$$C_3 = (v_1, \ldots, v_p, v_{p+1}^{'}, \ldots, v_N^{'}, g_1, \ldots, g_p, g_{p+1}^{'}, \ldots, g_N^{'})$$
$$C_4 = (v_1^{'}, \ldots, v_p^{'}, v_{p+1}, \ldots, v_N, g_1^{'}, \ldots, g_p^{'}, g_{p+1}, \ldots, g_N)$$

3.4.3. *Mutation*

Two different operators are used, each one of them acting on different chromosome parts. A brief description of them is given below:

- *Mutation on* C_V: As this part of the chromosome is binary coded, a simple binary mutation is developed, flipping the value of the gene.
- *Mutation on* C_G: The mutation operator selected for C_G performs a slight change in the selected variable. Once a granularity level is randomly selected to be muted, a local modification is developed by changing the number of labels of the variable to the immediately upper or lower value (the decision is made at random). When the value to be changed is the lowest (2) or highest one (7), the only possible change is developed.

4. Experimental Study

In this section, we will first provide details of the imbalanced problems chosen for the experimentation (Subsec. 4.1). Then, we will introduce the algorithms selected for comparison and the configuration parameters (Subsec. 4.2). Next, we will describe the statistical tests applied to compare the results obtained along the experimental study (Subsec. 4.3). Finally, we show the results obtained for all the methods and the statistical analysis (Subsec. 4.4).

4.1. *Data-sets*

We will study the performance of GA-FS+GL employing a large collection of imbalanced data-sets with high IR, considering a threshold value of 9 (distribution 1:10). Specifically, we have considered twenty-two data-sets from KEEL data-set repository[33] with different IR, as shown in Table 2, where we denote 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 in ascendant order

Table 2. Summary description for imbalanced data-sets.

Data-set	#Ex.	#Atts.	Class (min., maj.)	%Class (min.; maj.)	IR
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	(vac; nuc)	(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

according to the IR. Multi-class data-sets are modified to obtain two-class imbalanced problems, defining the joint of one or more classes as positive and the joint of one or more classes as negative. In order to reduce the effect of imbalance, we will employ the SMOTE preprocessing method[34] for all our experiments, considering only the 1-nearest neighbor to generate the synthetic samples, and balancing both classes to the 50% distribution.

We have obtained the *AUC* metric estimates by means of a 5-fold cross-validation. That is, the data-set was split into 5 folds, each one containing 20% of the patterns of the data-set. For each fold, the algorithm is trained with the examples contained in the remaining folds and then tested with the current fold. The data partitions used in this paper can be found in KEEL data-set repository[33] (`http://www.keel.es/dataset.php`), both for the original partitions and those preprocessed data with the SMOTE method, so that any interested researcher can reproduce the experimental study.

Finally, since a GA is a probabilistic method, three runs with different seeds for the pseudo-random sequence are made for each data partition. For each data-set we consider the average results of the five partitions per three executions.

4.2. *Algorithms of comparison and parameters*

We will analyze the influence of feature selection and granularity learning by means of a comparison between the performance of GA-FS+GL and three other methods with all the variables selected:

- The original Chi *et al.*'s method,[27] that needs of the existence of a previous definition for the DB, normally uniform fuzzy partitions with the same number of labels in all the variables. Therefore, it is necessary to choose a number of labels, being the usual values employed for any standard FRBCS approach in the specialized literature 3, 5 and 7 labels per variable. According to this fact, we include these three possibilities in the experimental study. In the latter, we will refer these methods as G3-Chi, G5-Chi and G7-Chi.
- The method proposed in Ref. 18 (denoted GA-GL), that uses a GA (similar to the used in GA-FS+GL) for granularity learning and the Chi *et al.*'s method to derive the RB.
- C4.5,[28] a method of reference in the field of classification with imbalanced data-sets.[29–32]

The configuration for the FRBCSs approaches, GA-FS+GL, GA-GL, G3-Chi, G5-Chi and G7-Chi is presented in Table 3 being "Conjuction operator" the operator used to compute the compatibility degree of the example with the antecedent of the rule and the operator used to compute the compatibility degree and the rule weight. This parameter selection has been carried out according to the results achieved by the Chi *et al.*'s method in our former studies on imbalanced data-sets:[12]

Table 3. Configuration for the FRBCS.

Conjunction operator:	Product T-norm
Rule Weight:	Penalized Certainty Factor[38]
Fuzzy Reasoning Method:	Winning Rule

The specific parameters setting for the GA of GA-FS+GL is listed below, being N the number of variables:

- Number of evaluations: $500 \cdot N$
- Population Size: 100 individuals
- Crossover Probability P_c : 0.6
- Mutation Probability P_m : 0.2
- Fitness function weights: (ω_1 : 0.5, α_{ω_2} : 0.5)

The most important parameters are the weighting factors of the evaluation function, that determining whether GA-FS+GL looks for more accurate solutions or less complex solutions. We have tested several values for ω_1 $(1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3)$ with their correspondent values for α_{ω_2} (α_{ω_2} =

$1 - \omega_1$). Since the feature selection ensures a significant reduction of the complexity, we have chosen the values shown before as they obtain the best mean for the AUC over the test data-set (high prediction ability). In the appendix, we show the obtained results for all tested values of ω_1 (Table 9).

4.3. *Statistical tests for performance comparison*

In this paper, we use the hypothesis testing techniques to provide statistical support to the analysis of the results.[37,59] Specifically, we will use non-parametric tests, due to the fact that the initial conditions that guarantee the reliability of the parametric tests may not be satisfied, making the statistical analysis to lose credibility with these type of tests.[35]

In a first approach, we apply the Wilcoxon signed-rank test[59] as non-parametric statistical procedure for performing pairwise comparisons between two algorithms. We will also compute the p-value associated to each comparison, which represents the lowest level of significance of a hypothesis that results in a rejection. In this manner, we can know whether two algorithms are significantly different and how different they are.

Additionally, since this test cannot assume the symmetry in the population of differences, to contrast our hypothesis we also perform a single significance test for every pair of algorithms using a sign test[59] on the win/draw/loss record of the two algorithms across all data-sets. Specifically, this test does not assume any commensurability of scores or differences nor does it assume normal distributions and is thus applicable to any data.[35] However, it has a lower asymptotic relative efficiency than the Wilcoxon signed-ranks test and therefore it has an inferior power.

Furthermore, we consider the average ranking of the algorithms in order to show graphically how good a method is with respect to its partners. This ranking is obtained by assigning a position to each algorithm depending on its performance for each data-set. The algorithm which achieves the best accuracy on a specific data-set will have the first ranking (value 1); then, the algorithm with the second best accuracy is assigned rank 2, and so forth. This task is carried out for all data-sets and finally an average ranking is computed as the mean value of all rankings.

These tests are suggested in the studies presented in,[35–37,60] where its use in the field of machine learning is highly recommended. Any interested reader can find additional information on the Website `http://sci2s.ugr.es/sicidm/`, together with the software for applying the statistical tests.

4.4. *Experimental analysis*

Table 4 shows the results in performance (using the AUC metric) for GA-FS+GL and the algorithms employed for comparison, that is, G3-Chi, G5-Chi, G7-Chi, GA-GL and C4.5, being Tr the AUC over the training data-set and Tst the AUC over the test data-set.

Table 4. Detailed table of results for the different methods in Train (Tr) and Test (Tst).

Data-set	G3-Chi		G5-Chi		G7-Chi		GA-GL		GA-FS+GL		C4.5	
	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
Yeast2vs4	.8968	.8736	.9051	.8685	.9695	.8510	.9391	.8985	.9004	.8840	.9814	.8588
Yeast05679vs4	.8265	.7917	.8797	.7642	.9295	.7226	.8640	.8296	.8147	.7928	.9526	.7602
Vowel0	.9857	.9839	.9964	.9789	.9939	.9371	.9955	.9917	.9501	.9282	.9967	.9494
Glass016vs2	.6271	.5417	.7616	.6002	.8603	.5429	.8630	.6360	.6916	.6262	.9716	.6062
Glass2	.6654	.5530	.7550	.5206	.8767	.5684	.8382	.5000	.7083	.7285	.9571	.5424
Ecoli4	.9406	.9151	.9814	.9230	.9833	.8151	.9810	.9183	.8838	.8595	.9769	.8310
shuttle0vs4	1.0000	.9912	1.0000	.9872	1.0000	.9870	1.0000	.9915	.9999	.9994	.9999	.9997
yeastB1vs7	.8200	.8063	.8408	.6524	.9124	.6648	.8148	.7753	.7611	.7544	.9351	.7003
Glass4	.9527	.8570	.9888	.8285	.9961	.7635	.9869	.8435	.9004	.8530	.9844	.8508
Page-Blocks13vs4	.9368	.9205	.9871	.9341	.9986	.8819	.9961	.9899	.9493	.9369	.9975	.9955
Abalone9-18	.7023	.6470	.7122	.6744	.7949	.6894	.8177	.7121	.6658	.6275	.9531	.6215
Glass016vs5	.9057	.7971	.9843	.8486	.9858	.6757	.9857	.8514	.9121	.8671	.9921	.8129
shuttle2vs4	.9500	.9078	1.0000	.8838	1.0000	.8838	.9970	.9878	.9959	.9920	.9990	.9917
Yeast1458vs7	.7125	.6465	.8183	.5932	.8848	.6310	.8510	.6522	.6729	.6406	.9158	.5367
Glass5	.9433	.8317	.9878	.7463	.9894	.6744	.9811	.8134	.9071	.7549	.9976	.8829
Yeast2vs8	.7861	.7728	.8346	.8066	.9331	.7055	.8454	.7946	.7809	.7104	.9125	.8066
Yeast4	.8358	.8315	.8796	.8325	.9074	.7855	.8742	.8078	.8382	.8346	.9101	.7004
Yeast1289vs7	.7470	.7712	.8003	.7027	.8609	.6139	.8113	.6716	.7343	.7499	.9465	.6832
Yeast5	.9468	.9358	.9543	.9372	.9782	.9413	.9643	.9493	.9559	.9354	.9777	.9233
Yeast6	.8848	.8809	.8960	.8820	.9536	.8851	.9125	.8698	.8974	.8698	.9242	.8280
Ecoli0137vs26	.9396	.8190	.9685	.6880	.9812	.6333	.9763	.8136	.9012	.7900	.9678	.8136
Abalone19	.7144	.6394	.7719	.6748	.8160	.6251	.8014	.6779	.6843	.6991	.8544	.5202
Mean	.8509	.8052	.8956	.7876	.9366	.7490	.9135	.8171	.8412	.8107	.9593	.7825

As it can be observed, the prediction ability obtained by GA-FS+GL is higher than the obtained for the other methods, except for GA-GL, showing the significative influence of the joining of feature selection and granularity level in the behavior of the classifier regarding to the classical way to proceed. A GA designed only for granularity learning (GA-GL) obtains the best results in prediction ability but with an increase of the model complexity. The number of rules is a typical measure used to compare the complexity of the models. Table 5 show the average number of rules obtained with each method. The number of rules of GA-FS+GL is always lower by the feature selection process and very much lower compared with the other FRBCSs, reducing the complexity of the model. The reduction in the number of rules is about 60% compared with C4.5, about 87% compared with G3-Chi, almost 90% compared with GA-GL and greater than 92% in the other methods (G5-Chi and G7-Chi) Therefore, the interpretability of the FRBCSs generated by GA-FS+GL is much higher. GA-FS+GL obtains FRBCSs with similar prediction ability than GA-GL but a great improvement in interpretability. In the appendix, we show the average number of rules for all tested values of ω_1 (Table 10) and another table with the average run time of all the approaches compared in this paper (Table 11).

Table 5. Average number of rules for the different data-sets.

Data-set	G3-Chi	G5-Chi	G7-Chi	GA-GL	GA-FS+GL	C4.5
Yeast2vs4	43.00	164.80	246.80	45.00	2.20	20.40
Yeast05679vs4	63.40	191.20	343.60	78.80	9.40	30.00
Vowel0	323.20	694.60	798.80	396.60	4.40	11.80
Glass016vs2	32.60	65.20	111.00	70.00	3.60	15.20
Glass2	33.20	73.20	110.80	70.00	8.20	16.80
Ecoli4	46.80	116.80	196.00	42.80	58.20	9.20
shuttle0vs4	25.80	79.20	78.00	22.40	14.80	2.80
yeastB1vs7	70.80	156.80	323.40	70.60	6.80	33.20
Glass4	42.20	98.40	155.00	50.60	8.40	7.40
Page-Blocks13vs4	64.80	164.00	256.40	81.00	6.60	7.00
Abalone9-18	43.40	93.60	134.60	57.20	4.40	47.60
Glass016vs5	48.00	99.60	152.00	55.40	4.00	10.00
shuttle2vs4	11.00	33.00	46.40	15.20	4.20	4.00
Yeast1458vs7	80.20	181.00	401.00	169.20	3.80	47.80
Glass5	41.60	90.80	137.00	43.60	5.20	7.00
Yeast2vs8	40.80	110.00	190.00	39.40	5.00	14.60
Yeast4	86.20	197.20	446.80	61.60	6.40	40.40
Yeast1289vs7	78.00	160.80	350.20	100.80	13.80	58.40
Yeast5	101.40	206.60	480.80	73.60	4.80	11.60
Yeast6	87.40	198.80	434.60	54.20	5.00	21.60
Ecoli0137vs26	77.80	168.60	245.00	76.60	4.20	8.00
Abalone19	69.20	180.20	346.20	90.60	7.40	69.20
Mean	68.67	160.20	272.02	80.24	8.67	22.45

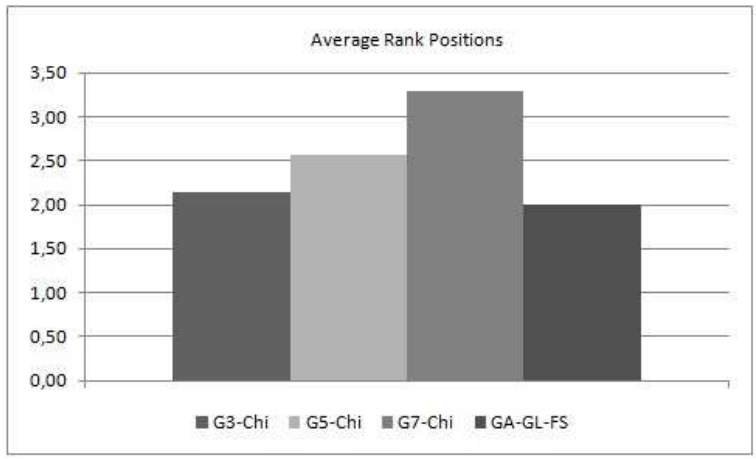


Fig. 5. Ranking for the GA-FS+GL approach and the Chi *et al.*'s method with 3, 5 and 7 labels.

In order to validate these results, we show the ranking on precision of the different models by means of the procedure described in Subsec. 4.3. Figure 5 shows the average ranking computed for the three different fuzzy alternatives: the three basic chi approaches with 3, 5 and 7 labels, and our GA-FS+GL proposal.

Next, we perform a sign test and a Wilcoxon test for detecting significant differences between the results of GA+FS+GL and the standard FRBCS approach with 3, 5 and 7 labels. The results of these tests are shown in Table 6 where, by columns, it is represented the current comparison, the number of wins, ties and loses for the GA-FS+GL approach versus the FRBCS, the sum of the ranks for GA-FS+GL and the FRBCS respectively, and the p-values obtained, first by the sign test, and second by the Wilcoxon test.

Specifically, in this table we observe that our GA-FS+GL model outperforms the basic Chi *et al.*'s method for the ones with the highest granularity levels (5 and 7 fuzzy partitions) with a low p-value in both cases which, in other words, implies that we can state, with a high degree of confidence, that our methodology is statistically superior to Chi-5 and Chi-7. Regarding the remaining approach, we may stress that the improvement of our proposal is not focused on the accuracy performance but on the high interpretability, as we will discuss below.

Table 6. Sign and Wilcoxon tests to compare GA-FS+GL [R^+] with the FRBCSs (Chi with 3, 5 and 7 labels) [R^-] regarding the AUC metric.

Comparison	w/t/l	R^+	R^-	Sign p-value	Wcx p-value
GA-FS+GL vs. Chi3	10/0/12	124.0	129.0	0.738	0.935
GA-FS+GL vs. Chi5	16/0/6	182.0	71.0	0.026	0.072
GA-FS+GL vs. Chi7	18/0/4	231.0	22.0	0.002	0.001

Table 7. Sign and Wilcoxon tests to compare GA-FS+GL [R^+] with GA-GL and C4.5 [R^-] regarding the AUC metric.

Comparison	w/t/l	R^+	R^-	Sign p-value	Wcx p-value
GA-FS+GL vs. GA-GL	13/0/9	84.0	169.0	0.262	0.168
GA-FS+GL vs. C4.5	16/0/6	186.0	67.0	0.026	0.053

In order to contrast the behaviour of GA-FS+GL with GA-GL (without feature selection) and C4.5, we carry out again a sign test and Wilcoxon test (Table 7) in which R^+ corresponds to the sum of the ranks for the GA-FS+GL approach and R^- to GA-GL and C4.5 respectively. We observe a low p-value in both cases, especially in the case of the comparison with C4.5, which allow us to determine the good results of our approach with the support of statistical differences in favour of our methodology. Furthermore, and as we have discussed before, the high level of interpretability of the classification models generated with our methodology, according to the low number of rules, the few number of antecedents/variables in these rules, and the use of a linguistic approach, derives in a higher degree of usefulness of our proposed approach with respect to the algorithms used for comparison, namely Chi *et al.*'s, GA-GL and C4.5.

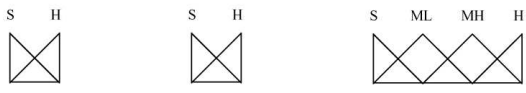
In fact, GA-FS+GL obtains precise and interpretable models by selecting a reduced set of features and finding an appropriate granularity level in each selected variable. Thus, we show in Table 8 the mean of selected variables (SV) in the first column. The remaining columns show two values for each feature of the problem, the first is the selection ratio of the variable, that is, the relation between the number of occasions in that the variable was selected and the number of total executions for each problem. The second value is the average of the number of labels for the cases in which that variable was selected.

As it can be observed in Table 8, the number of selected variables is very low. In all the problems the number of selected features is reduced, at least, to the half of the original (with only one exception, the data-set *"Yeast2vs8"*). Moreover, in 18 problems, the number of selected variables in the average of the 15 executions is less or equal than three deriving in a significant reduction in the length of the antecedent of the rules as point before. Regarding to the granularity level mean, there are significant differences among the variables of each data-set. This situation is caused by the advantage of increasing or decreasing the granularity for a good data representation in the fuzzy partition. Therefore, GA-FS+GL obtain FRBCSs with high prediction ability and very reduced complexity, that was the main purpose of this approach. Figures 6 and 7 show a representation of the KB obtained for GA-FS+GL, GA-GL and C4.5 for the first partition of the Glass2 data-set illustrating the high reduction of complexity of the models generated by GA-FS+GL.

Table 8. Mean of number of variables and labels per variable learned by GA-FS+GL.

Data-set	SV	1	2	3	4	5	6	7	8	9	10	11	12	13
Yeast2vs4	3.0	1./2.0	.6/2.0	1./3.0	.0/0.0	.0/0.0	.0/0.0	.2/2.0	.2/2.0	x	x	x	x	x
Yeast05679vs4	2.2	1./3.0	.2/3.0	.6/2.0	.0/0.0	.0/0.0	.0/0.0	.2/3.0	.2/2.0	x	x	x	x	x
Vowel0	5.2	.6/2.3	.6/3.7	.4/2.0	.8/4.0	1./4.6	.2/6.0	.6/3.0	.2/2.0	.2/2.0	.0/0.0	.0/0.0	.2/2.0	.4/2.0
Glass016vs2	3.0	.8/2.8	.6/2.3	.6/2.0	.0/0.0	.4/3.5	.0/0.0	.0/0.0	.2/2.0	.4/2.0	x	x	x	x
Glass2	3.0	.6/2.0	.4/2.0	1./2.0	.0/0.0	1./3.6	.0/0.0	.0/0.0	.0/0.0	.0/0.0	x	x	x	x
Ecoli4	2.8	1./2.0	1./2.4	.0/0.0	.0/0.0	.0/0.0	.2/2.0	.6/2.0	x	x	x	x	x	x
shuttle0vs4	2.2	1./3.0	.2/3.0	.2/3.0	.0/0.0	.0/0.0	.0/0.0	.8/2.0	.0/0.0	.0/0.0	x	x	x	x
yeast1vs7	2.8	1./2.2	.0/0.0	.8/2.0	.6/2.0	.0/0.0	.2/2.0	.2/2.0	x	x	x	x	x	x
Glass4	3.2	.6/2.0	.8/3.0	1./2.4	.0/0.0	.4/2.5	.0/0.0	.0/0.0	.4/2.0	.0/0.0	x	x	x	x
Page-Blocks13vs4	3.2	1./4.8	.8/3.0	.0/0.0	.0/0.0	.4/3.5	.4/2.0	.0/0.0	.0/0.0	.4/2.0	.2/2.0	x	x	x
Abalone9-18	2.8	.0/0.0	.6/2.7	.6/2.3	.2/2.0	.0/0.0	.4/2.0	.6/2.0	.4/3.0	x	x	x	x	x
Glass016vs5	3.0	.4/2.0	.8/2.8	1./2.8	.0/0.0	.2/2.0	.0/0.0	.2/3.0	.4/2.0	.0/0.0	x	x	x	x
shuttle2vs4	3.0	1./3.2	.0/0.0	1./2.8	.0/0.0	.0/0.0	.0/0.0	.8/2.0	.0/0.0	.2/2.0	x	x	x	x
Yeast1458vs7	2.2	1./4.2	.0/0.0	.0/0.0	.0/0.0	.0/0.0	.0/0.0	.2/2.0	1./2.0	x	x	x	x	x
Glass5	3.0	.8/2.5	.6/3.0	1./3.0	.2/2.0	.2/2.0	.0/0.0	.2/2.0	.0/0.0	.0/0.0	x	x	x	x
Yeast2vs8	5.4	.8/2.0	.8/3.0	.6/2.0	.8/2.8	.8/2.5	.8/2.3	.4/2.5	.4/3.0	x	x	x	x	x
Yeast4	2.8	1./3.0	.8/2.0	.8/2.0	.0/0.0	.0/0.0	.0/0.0	.0/0.0	.2/2.0	x	x	x	x	x
Yeast1289vs7	3.0	1./2.0	.0/0.0	1./3.0	.0/0.0	.0/0.0	.0/0.0	.6/2.0	.4/2.0	x	x	x	x	x
Yeast5	3.0	1./2.8	.6/2.3	.8/2.0	.6/2.0	.0/0.0	.0/0.0	.0/0.0	.0/0.0	x	x	x	x	x
Yeast6	2.8	.8/3.0	1./2.0	.2/3.0	.0/0.0	.0/0.0	.0/0.0	.0/0.0	.8/2.0	x	x	x	x	x
Ecoli0137vs26	2.4	1./2.6	.2/2.0	.4/2.0	.0/0.0	.0/0.0	.6/2.0	.2/2.0	x	x	x	x	x	x
Abalone19	2.0	1./2.6	1./4.0	.0/0.0	.0/0.0	.0/0.0	.0/0.0	.0/0.0	.0/0.0	x	x	x	x	x

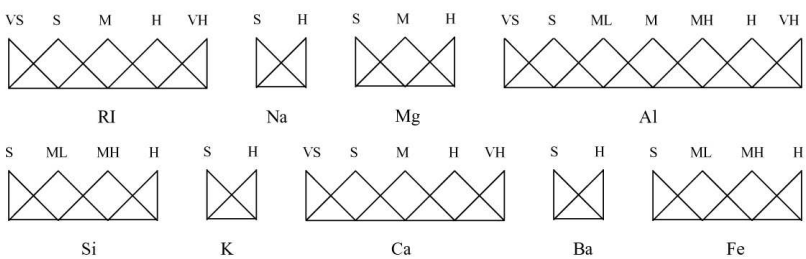
GA-FS+GL



R1: If Na is S and Mg is S and Si is ML Then Class = negative with RW = 0.10
R2: If Na is H and Mg is S and Si is MH Then Class = negative with RW = 0.50
R3: If Na is S and Mg is H and Si is MH Then Class = positive with RW = 0.01
R4: If Na is H and Mg is H and Si is ML Then Class = positive with RW = 0.27

VS: Very Small S: Small ML: Medium Low M: Medium MH: Medium High H: High VH:Very High

GA-GL



R1: If RI is S and Na is S and Mg is H and Al is ML and Si is MH and K is S and Ca is S and Ba is S and Fe is ML Then Class = negative with RW = 0.37
R2: If RI is M and Na is S and Mg is H and Al is S and Si is ML and K is S and Ca is M and Ba is S and Fe is ML Then Class = negative with RW = 0.53
R3: If RI is S and Na is S and Mg is H and Al is M and Si is MH and K is S and Ca is S and Ba is S and Fe is S Then Class = negative with RW = 0.87
R4: If RI is S and Na is S and Mg is M and Al is ML and Si is MH and K is S and Ca is S and Ba is S and Fe is ML Then Class = negative with RW = 0.45
R5: If RI is M and Na is S and Mg is S and Al is M and Si is ML and K is S and Ca is H and Ba is S and Fe is MH Then Class = negative with RW = 1.0
R6: If RI is S and Na is S and Mg is H and Al is M and Si is ML and K is S and Ca is S and Ba is S and Fe is S Then Class = negative with RW = 0.58
R7: If RI is S and Na is H and Mg is M and Al is M and Si is ML and K is S and Ca is S and Ba is S and Fe is S Then Class = negative with RW = 0.57
R8: If RI is S and Na is H and Mg is S and Al is M and Si is MH and K is S and Ca is S and Ba is S and Fe is S Then Class = negative with RW = 1.0
R9: If RI is M and Na is H and Mg is S and Al is S and Si is MH and K is S and Ca is M and Ba is S and Fe is S Then Class = negative with RW = 1.0
R10: If RI is H and Na is S and Mg is S and Al is S and Si is MH and K is S and Ca is VH and Ba is S and Fe is S Then Class = negative with RW = 1.0
R11: If RI is S and Na is S and Mg is H and Al is ML and Si is ML and K is S and Ca is S and Ba is S and Fe is ML Then Class = negative with RW = 0.24

……

Other 58 rules with the same structure

……

R70: If RI is M and Na is S and Mg is H and Al is VS and Si is ML and K is S and Ca is M and Ba is S and Fe is S Then Class = positive with RW = 0.51

Fig. 6. Knowledge Bases for Glass2 dataset obtained by GA-FS+GL and GA-GL.

C45

```
if ( Mg <= 3.350000 ) then  class = "negative"
 elseif ( Mg > 3.350000 ) then
  {
   if ( Ca <= 8.240000 ) then
    {
     if ( Mg <= 3.536547 ) then
      {
       if ( Ca <= 8.090000 )  then   class = "negative"
        elseif ( Ca > 8.090000 ) then  {
            if ( Na <= 13.030103 ) then   class = "negative"
            elseif ( Na > 13.030103 ) then   class = "positive"  }
       }
      elseif ( Mg > 3.536547 ) then   class = "negative"
    }
   elseif ( Ca > 8.240000 ) then
    {
     if ( RI <= 1.517162 ) then   class = "positive"
      elseif ( RI > 1.517162 ) then
       {
        if ( Na <= 13.310439 ) then
         {
          if ( Ba <= 0.090000 ) then
           {
            if ( Si <= 72.922681 ) then
             {
              if ( Na <= 12.923964 ) then      class = "positive"
               elseif ( Na > 12.923964 ) then  {
                  if ( Al <= 1.696739 ) then   class = "negative"
                  elseif ( Al > 1.696739 ) then  class = "positive"  }
              elseif ( Si > 72.922681 ) then   class = "negative"
             }
            elseif ( Ba > 0.090000 ) then  class = "positive"
           }
          elseif ( Na > 13.310439 ) then
           {
            if ( RI <= 1.518410 ) then
             {
              if ( Ca <= 8.430000 ) then  {
                 if ( RI <= 1.517680 ) then   class = "negative"
                 elseif ( RI > 1.517680 ) then   class = "positive"   }
               elseif ( Ca > 8.430000 ) then   class = "positive"
             }
            elseif ( RI > 1.518410 ) then
             {
              if ( RI <= 1.522110 ) then
               {
                if ( Al <= 1.092882 ) then   class = "positive"
                 elseif ( Al > 1.092882 ) then  {
                     if ( Al <= 1.450000 ) then  class = "negative"
                     elseif ( Al > 1.450000 ) then  class = "positive"  }
               }
              elseif ( RI > 1.522110 ) then          class = "negative"
             }
           }
         }
       }
    }
  }
```

Fig. 7. Knowledge Bases for Glass2 dataset obtained by C4.5.

5. Conclusions

This contribution has described a methodology to design linguistic FRBCSs with good accuracy and very reduced complexity for highly imbalanced data-sets. A GA is used for feature selection and granularity learning, which is combined with an efficient fuzzy classification rule generation method to obtain the complete KB of the FRBCS.

Our proposal is compared with various classical and modern methods, obtaining similar or better results in prediction ability with always a significant enhancement in the interpretability of the model. The improvement in the interpretability is due to the high reduction on the number of rules of the model and the fact that these rules are simpler as they use less variables in their antecedent part.

We must remark that one advantage of our proposal is that the GA can be combined with any rule generation method. We have used a simple algorithm for efficiency but more accurate ones can be used, or another more suitable for a specific data-set.

Acknowledgments

This work had been supported by the Spanish Ministry of Science and Technology under Project TIN2008-06681-C06-01.

References

1. N. V. Chawla, N. Japkowicz, and A. Kolcz, Editorial: special issue on learning from imbalanced data sets, *SIGKDD Explorations*, vol. 6, no. 1, pp. 1–6, 2004.
2. H. He and E. A. Garcia, Learning from imbalanced data, *IEEE Transactions On Knowledge And Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.
3. Y. Sun, A. K. C. Wong, and M. S. Kamel, Classification of imbalanced data: A review, *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687–719, 2009.
4. M. Mazurowski, P. Habas, J. Zurada, J. Lo, J. Baker, and G. Tourassi, Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance, *Neural Networks*, vol. 21, no. 2–3, pp. 427–436, 2008.
5. X. Peng and I. King, Robust BMPM training based on second–order cone programming and its application in medical diagnosis, *Neural Networks*, vol. 21, no. 2–3, pp. 450–457, 2008.
6. Y. M. Huang, C. M. Hung, and 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*, vol. 7, no. 4, pp. 720–747, 2006.
7. Y.-H. Liu and Y.-T. Chen, Face recognition using total margin–based adaptive fuzzy support vector machines, *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 178–192, 2007.
8. D. Williams, V. Myers, and M. Silvious, Mine classification with imbalanced data, *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 3, pp. 528–532, 2009.
9. C.-H. Tsai, L.-C. Chang, and H.-C. Chiang, Forecasting of ozone episode days by cost–sensitive neural network methods, *Science of the Total Environment*, vol. 407, no. 6, pp. 2124–2135, 2009.

10. W.-Z. Lu and D. Wang, Ground–level ozone prediction by support vector machine approach with a cost–sensitive classification scheme, *Science of the Total Environment*, vol. 395, no. 2–3, pp. 109–116, 2008.
11. H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining.* Springer–Verlag, 2004.
12. A. Fernández, S. García, M. J. del Jesus, and 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*, vol. 159, no. 18, pp. 2378–2398, 2008.
13. O. Cordón, F. Herrera, and P. Villar, Analysis and guidelines to obtain a good uniform fuzzy partition granularity for fuzzy rule–based systems using simulated annealing, *International Journal of Approximate Reasoning*, vol. 25, no. 3, pp. 187–215, 2000.
14. ——, Generating the knowledge base of a fuzzy rule–based system by the genetic learning of the data base, *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 667–674, 2001.
15. O. Cordón, F. Herrera, L. Magdalena, and P. Villar, A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule–based system data base, *Information Sciences*, vol. 136, pp. 85–107, 2001.
16. E. Zhou and A. Khotanzad, Fuzzy classifier design using genetic algorithms, *Pattern Recognition*, vol. 40, no. 12, pp. 3401–3414, 2007.
17. I. Walter and F. Gomide, Genetic fuzzy systems to evolve interaction strategies in multiagent systems, *International Journal of Intelligent Systems*, vol. 22, no. 9, pp. 971–991, 2007.
18. P. Villar, A. Fernández, and F. Herrera, A genetic learning of the fuzzy rule–based classification system granularity for highly imbalanced data–sets, in *2009 IEEE International Conference on Fuzzy Systems (FUZZ–IEEE09)*, 2009, pp. 1689–1694.
19. S. Halgamuge and M. Glesner, Neural networks in designing fuzzy systems for real world applications, *Fuzzy Sets and Systems*, vol. 65, no. 1, pp. 1–12, 1994.
20. S. Chiu, Fuzzy model identification based on cluster estimation, *Journal of Intelligent & fuzzy systems*, vol. 2, pp. 267–278, 1994.
21. J. Yen and L. Wang, Simplifying fuzzy rule-based models using orthogonal transformation methods, *IEEE Transactions on System, Man and Cybernetics B*, vol. 29, no. 1, pp. 13–24, 1999.
22. M. Setnes, R. Babuska, U. Kaymak, and H. van Nauta-Lemke, Similariry measures in fuzzy rule base simplification, *IEEE Transactions on System, Man and Cybernetics B*, vol. 28, no. 3, pp. 376–386, 1998.
23. Y. Jin, Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement, *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 2, pp. 212–221, 2000.
24. H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, Selecting fuzzy if–then rules for classification problems using genetic algorithms, *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 3, pp. 260–270, 1995.
25. O. Cordón and F. Herrera, A three–stage evolutionary process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples, *International Journal of Approximate Reasoning*, vol. 17, no. 4, pp. 369–407, 1997.
26. J. Casillas, O. Cordón, M. J. del Jesus, and F. Herrera, Genetic feature selection in a fuzzy rule–based classification system learning process for high dimensional problems, *Information Sciences*, vol. 136, pp. 135–157, 2001.
27. Z. Chi, H. Yan, and T. Pham, *Fuzzy algorithms with applications to image processing and pattern recognition.* World Scientific, 1996.

28. J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo–California: Morgan Kaufmann Publishers, 1993.
29. G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, A study of the behaviour of several methods for balancing machine learning training data, *SIGKDD Explorations*, vol. 6, no. 1, pp. 20–29, 2004.
30. A. Orriols-Puig and E. Bernadó-Mansilla, Evolutionary rule–based systems for imbalanced datasets, *Soft Computing*, vol. 13, no. 3, pp. 213–225, 2009.
31. C.-T. Su, L.-S. Chen, and Y. Yih, Knowledge acquisition through information granulation for imbalanced data, *Expert Systems with Applications*, vol. 31, pp. 531–541, 2006.
32. C.-T. Su and Y.-H. Hsiao, An evaluation of the robustness of MTS for imbalanced data, *IEEE Transactions on Knowledge Data Engineering*, vol. 19, no. 10, pp. 1321–1332, 2007.
33. J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework, *Journal of Multi-Valued Logic and Soft Computing*, vol. in press, 2010.
34. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, SMOTE: Synthetic minority over–sampling technique, *Journal of Artificial Intelligent Research*, vol. 16, pp. 321–357, 2002.
35. J. Demšar, Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
36. S. García and F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, *Journal of Machine Learning Research*, vol. 9, pp. 2677–2694, 2008.
37. S. García, A. Fernández, J. Luengo, and F. Herrera, A study of statistical techniques and performance measures for genetics–based machine learning: Accuracy and interpretability, *Soft Computing*, vol. 13, no. 10, pp. 959–977, 2009.
38. H. Ishibuchi and T. Yamamoto, Rule weight specification in fuzzy rule–based classification systems, *IEEE Transactions on Fuzzy Systems*, vol. 13, pp. 428–435, 2005.
39. A. Fernández, M. J. del Jesus, and F. Herrera, Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data–sets, *International Journal of Approximate Reasoning*, vol. 50, pp. 561–577, 2009.
40. ——, On the influence of an adaptive inference system in fuzzy rule based classification systems for imbalanced data–sets, *Expert Systems With Applications*, vol. 36, no. 6, pp. 9805–9812, 2009.
41. L. X. Wang and J. M. Mendel, Generating fuzzy rules by learning from examples, *IEEE Transactions on System, Man and Cybernetics*, vol. 25, no. 2, pp. 353–361, 1992.
42. G. Weiss and F. Provost, Learning when training data are costly: The effect of class distribution on tree induction, *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.
43. A. Orriols-Puig, E. Bernadó-Mansilla, D. E. Goldberg, K. Sastry, and P. L. Lanzi, Facetwise analysis of XCS for problems with class imbalances, *IEEE Transactions on Evolutionary Computation*, vol. 13, pp. 260–283, 2009.
44. V. García, R. Mollineda, and J. S. Sánchez, On the k-NN performance in a challenging scenario of imbalance and overlapping, *Pattern Analysis Applications*, vol. 11, no. 3–4, pp. 269–280, 2008.
45. R. Barandela, J. S. Sánchez, V. García, and E. Rangel, Strategies for learning in class imbalance problems, *Pattern Recognition*, vol. 36, no. 3, pp. 849–851, 2003.

46. M.-C. Chen, L.-S. Chen, C.-C. Hsu, and W.-R. Zeng, An information granulation based data mining approach for classifying imbalanced data, *Information Sciences*, vol. 178, no. 16, pp. 3214–3227, 2008.
47. P. Ducange, B. Lazzerini, and F. Marcelloni, Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets, *Soft Computing*, vol. 14, no. 7, pp. 713–728, 2010.
48. G. Wu and E. Chang, KBA: Kernel boundary alignment considering imbalanced data distribution, *IEEE Transactions on Knowledge Data Engineering*, vol. 17, no. 6, pp. 786–795, 2005.
49. L. Xu, M. Chow, and L. Taylor, Power distribution fault cause identification with imbalanced data using the data mining-based fuzzy classification e-algorithm, *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 164–171, 2007.
50. A. Estabrooks, T. Jo, and N. Japkowicz, A multiple resampling method for learning from imbalanced data sets, *Computational Intelligence*, vol. 20, no. 1, pp. 18–36, 2004.
51. 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.
52. Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, Cost–sensitive boosting for classification of imbalanced data, *Pattern Recognition*, vol. 40, pp. 3358–3378, 2007.
53. Z.-H. Zhou and X.-Y. Liu, Training cost–sensitive neural networks with methods addressing the class imbalance problem, *IEEE Transactions on Knowledge Data Engineering*, vol. 18, no. 1, pp. 63–77, 2006.
54. A. Fernández, M. J. del Jesus, and F. Herrera, On the 2–tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data–sets, *Information Sciences*, vol. 180, no. 8, pp. 1268–1291, 2010.
55. A. P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.
56. J. Huang and C. X. Ling, Using AUC and accuracy in evaluating learning algorithms, *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, 2005.
57. J. A. Hanley and B. J. McNeil, The meaning and use of the area under a receiver operating characteristics (ROC) curve, *Radiology*, vol. 143, pp. 29–36, 1982.
58. D. Green and J. Swets, *Signal Detection Theory and Psychophysics*. New York: Wiley, 1966.
59. D. Sheskin, *Handbook of parametric and nonparametric statistical procedures*, 2nd ed. Chapman & Hall/CRC, 2006.
60. S. García, A. Fernández, J. Luengo, and F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, *Information Sciences*, vol. 180, pp. 2044–2064, 2010.

Appendix: Complementary tables

In this appendix we include three tables, with the detailed results of GA-FS+GL for all tested values of ω_1 for the AUC (Table 9) and the average of the number of rules (Table 10), apart from the table of the average run time of all the methods compared (Table 11).

Table 9. Detailed table of results for the different weight values for GA-FS+GL in Train (Tr) and Test (Tst).

Data-set	ω_1–α_{ω_2}															
	1.0–0.0		0.9–0.1		0.8–0.2		0.7–0.3		0.6–0.4		0.5–0.5		0.4–0.6		0.3–0.7	
	Acc_{Tr}	Acc_{Tst}	Acc_{Tr}	Acc_{Tst}	Acc_{Tr}	Acc_{Tst}	Acc_{Tr}	Acc_{Tst}	Acc_{Tr}	Acc_{Tst}	Acc_{Tr}	Acc_{Tst}	Acc_{Tr}	Acc_{Tst}	Acc_{Tr}	Acc_{Tst}
Yeast2vs4	97.08	87.52	93.90	92.48	91.61	90.22	90.04	88.40	89.55	88.73	90.04	88.40	89.68	88.62	89.55	89.40
Yeast05679vs4	93.38	74.52	91.94	81.59	82.91	80.32	81.57	79.59	81.54	79.49	81.47	79.28	81.66	79.07	81.44	78.65
Vowel0	99.81	97.55	99.29	97.94	99.12	97.72	99.05	96.77	96.16	96.33	95.01	92.82	90.05	90.99	92.57	88.10
Glass016vs2	89.79	58.33	88.64	60.83	84.02	61.69	81.37	58.26	73.68	58.83	69.16	62.62	67.87	60.60	68.12	59.55
Glass2	87.92	54.55	86.93	59.34	82.12	58.63	78.50	64.81	70.23	68.29	70.83	72.85	70.45	73.60	70.65	68.85
Ecoli4	99.17	84.50	97.21	89.49	97.86	94.80	95.93	91.79	90.43	85.67	88.38	85.95	87.30	84.72	88.38	86.43
Shuttle0vs4	100.00	99.17	99.99	99.91	99.98	99.91	99.99	99.94	99.97	99.94	99.99	99.94	99.98	99.10	99.98	99.12
Yeast1vs7	91.46	66.21	91.16	65.48	87.29	70.59	76.48	69.04	77.79	75.70	76.11	75.44	78.00	77.25	71.60	75.44
Glass4	99.88	86.17	99.13	88.43	98.45	91.76	98.38	91.75	95.73	91.52	90.04	85.30	91.59	88.77	89.35	86.56
Page-Blocks13vs4	100.00	94.98	99.52	98.53	99.38	96.65	99.10	99.21	97.57	97.53	94.93	93.69	94.53	95.83	94.75	94.25
Abalone9-18	84.29	70.00	83.24	69.41	81.23	69.65	71.98	62.39	67.00	63.93	66.58	62.75	66.15	60.70	66.57	61.97
Glass016vs5	99.14	73.71	98.29	79.86	97.21	79.00	93.57	87.00	92.93	88.43	91.21	86.71	92.07	87.86	93.29	89.57
Shuttle2vs4	100.00	88.38	99.39	89.60	99.39	88.75	99.80	100.00	99.39	89.17	99.59	99.20	99.59	99.20	99.70	99.17
Yeast1458vs7	89.00	61.20	88.91	61.35	85.68	67.71	84.02	65.59	68.53	59.61	67.29	64.06	67.25	60.98	67.30	62.34
Glass5	99.33	67.44	98.35	66.83	95.77	75.85	94.36	73.66	91.01	69.76	90.71	75.49	91.56	70.24	89.03	76.22
Yeast2vs8	92.70	77.51	86.44	78.59	85.63	79.24	82.50	76.74	83.15	79.03	78.09	71.04	63.20	56.18	64.40	53.55
Yeast4	91.72	80.85	90.71	81.48	85.20	81.90	83.79	83.32	84.03	83.08	83.82	83.46	84.03	83.08	84.03	83.08
Yeast1289vs7	89.38	65.38	88.31	61.77	82.77	65.27	74.38	74.07	73.44	75.05	73.43	74.99	74.54	74.72	72.86	70.14
Yeast5	97.98	92.29	95.83	92.78	95.66	93.33	95.58	93.51	95.60	93.72	95.59	93.54	95.61	93.75	95.70	93.44
Yeast6	95.28	86.00	89.28	86.15	89.43	87.01	89.36	86.32	89.62	86.43	89.74	86.98	89.74	86.98	89.54	86.74
Ecoli0137vs26	98.81	68.44	97.49	71.35	97.08	81.00	95.40	80.45	91.40	79.91	90.12	79.00	90.17	78.81	90.17	78.81
Abalone19	84.90	63.66	82.22	63.27	76.09	68.15	68.43	69.91	68.43	69.91	68.43	69.91	68.43	69.91	68.43	69.91
Mean	94.59	77.20	93.01	78.93	90.63	80.87	87.89	81.48	85.33	80.91	84.12	81.07	83.34	80.04	83.06	79.60

Table 10. Average number of rules for the different weight values of GA-FS+GL.

Data-set	ω_1–α_{ω_2}							
	1.0–0.0	0.9–0.1	0.8–0.2	0.7–0.3	0.6–0.4	0.5–0.5	0.4–0.6	0.3–0.7
Yeast2vs4	105.60	88.60	56.20	5.00	2.20	2.20	2.40	3.20
Yeast05679vs4	299.40	145.80	46.20	9.40	9.40	9.40	9.40	9.40
Vowel0	138.20	50.80	15.40	12.80	5.40	4.40	4.20	4.40
Glass016vs2	99.60	81.80	43.00	21.00	5.40	3.60	3.40	3.80
Glass2	345.40	130.40	15.00	8.20	8.00	8.20	8.00	8.00
Ecoli4	682.20	178.00	125.20	137.60	87.40	58.20	17.20	15.80
shuttle0vs4	142.60	66.40	38.40	14.40	14.20	14.80	4.60	5.00
yeastB1vs7	107.00	41.60	28.60	28.20	21.60	6.80	7.40	5.60
Glass4	116.40	45.80	23.40	19.40	7.80	8.40	7.00	7.00
Page-Blocks13vs4	428.20	13.20	6.40	7.00	7.60	6.60	7.20	6.40
Abalone9-18	353.40	9.00	5.00	5.40	5.60	4.40	4.40	5.80
Glass016vs5	180.60	44.00	27.00	20.00	7.60	4.00	3.60	3.60
shuttle2vs4	36.80	4.20	4.80	4.20	5.80	4.20	5.20	5.20
Yeast1458vs7	251.60	216.40	113.20	14.40	4.00	3.80	4.00	3.40
Glass5	33.60	12.00	8.00	6.80	7.80	5.20	5.20	5.00
Yeast2vs8	97.60	81.00	43.60	28.60	11.40	5.00	5.20	5.60
Yeast4	126.20	53.80	32.20	20.00	7.00	6.40	7.20	7.20
Yeast1289vs7	149.60	37.60	38.80	37.40	32.20	13.80	6.40	5.20
Yeast5	327.20	188.20	28.20	7.40	6.60	4.80	5.00	5.00
Yeast6	277.40	257.40	67.60	4.80	5.00	5.00	5.00	5.60
Ecoli0137vs26	306.20	310.60	150.40	137.20	12.80	4.20	5.20	5.40
Abalone19	241.00	28.80	13.20	7.40	6.60	7.40	7.40	6.60
Mean	220.26	94.79	42.26	25.30	12.79	8.67	6.12	6.01

Table 11. Average training time for the different data-sets.

Data-set	G3-Chi	G5-Chi	G7-Chi	GA-GL	GA-FS+GL	C4.5
Abalone9-18	00:00:04	00:00:04	00:00:04	00:36:40	00:15:51	00:00:00
Abalone19	00:00:40	00:00:39	00:00:39	19:54:53	08:51:38	00:00:00
Ecoli4	00:00:02	00:00:02	00:00:02	00:05:41	00:04:08	00:00:00
Glass2	00:00:01	00:00:01	00:00:01	00:04:18	00:04:09	00:00:00
Yeast4	00:00:07	00:00:07	00:00:08	02:24:21	01:04:02	00:00:00
Vowel0	00:00:07	00:00:07	00:00:07	02:56:59	01:22:35	00:00:00
Yeast2vs8	00:00:02	00:00:02	00:00:02	00:14:50	00:11:57	00:00:00
Glass4	00:00:01	00:00:01	00:00:01	00:04:00	00:04:19	00:00:00
Glass5	00:00:01	00:00:01	00:00:01	00:04:01	00:04:26	00:00:00
Yeast5	00:00:07	00:00:08	00:00:08	02:17:06	01:00:08	00:00:00
Yeast6	00:00:07	00:00:07	00:00:08	02:18:38	01:08:09	00:00:00
Ecoli0137vs26	00:00:01	00:00:01	00:00:01	00:04:35	00:03:31	00:00:00
shuttle0vs4	00:00:07	00:00:07	00:00:07	04:11:19	01:35:05	00:00:00
yeastB1vs7	00:00:02	00:00:02	00:00:02	00:11:34	00:06:10	00:00:00
shuttle2vs4	00:00:00	00:00:00	00:00:00	00:01:28	00:02:44	00:00:00
Glass016vs2	00:00:01	00:00:01	00:00:01	00:03:23	00:03:38	00:00:00
Glass016vs5	00:00:01	00:00:01	00:00:01	00:03:14	00:03:47	00:00:00
Page-Blocks13vs4	00:00:03	00:00:03	00:00:03	00:24:09	00:13:31	00:00:00
Yeast05679vs4	00:00:03	00:00:02	00:00:03	00:17:20	00:08:09	00:00:00
Yeast1289vs7	00:00:05	00:00:05	00:00:05	01:02:56	00:25:03	00:00:00
Yeast1458vs7	00:00:03	00:00:03	00:00:03	00:35:05	00:14:15	00:00:00
Yeast2vs4	00:00:02	00:00:02	00:00:02	00:15:30	00:08:07	00:00:00
Mean	00:00:05	00:00:05	00:00:05	01:44:11	00:47:04	00:00:00