2011 IEEE International Conference on Fuzzy Systems
June 27-30, 2011, Taipei, Taiwan

Studying the Behavior of a Multiobjective Genetic Algorithm to design Fuzzy Rule-Based Classification Systems for Imbalanced Data-Sets

Pedro Villar
Dept. of Software Engineering
University of Granada
Granada, Spain
Email: pvillarc@ugr.es

Alberto Fernández
Dept. of Computer Science
University of Jaén
Jaén, Spain
Email: alberto.fernandez@ujaen.es

Francisco Herrera
Dept. of Computer Science and Artificial Intelligence
University of Granada
Granada, Spain
Email: herrera@decsai.ugr.es

***Abstract*—This paper studies the behavior of a multiobjective Genetic Algorithm for jointly performing a feature selection and granularity learning for Fuzzy Rule-Based Classification Systems in the scenario of 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.**

We consider two different measures, one for the precision of the model and other for its complexity as the two objectives to optimize. In one previous approach, we aggregate these two measures in a single-objective Genetic Algorithm, and thus, a multiobjective approach of that Genetic Algorithm would yield a set of models with different trade-off between high accuracy and low complexity rather than a unique model, provided by the single-objective Genetic Algorithm. The experimental analysis, carried out over a wide range of imbalanced data-sets, shows that our approach is able to obtain a set of models with good trade-off between the two objectives considered but it is an open problem how to select the solution with best prediction ability from the whole set of solutions obtained.

***Index Terms*—Fuzzy Rule-Based Classification Systems, imbalanced data-sets, Multiobjective Genetic Algorithms, feature selection, granularity level.**

I. Introduction

Fuzzy Rule Based Classification Systems (FRBCSs) are considered a very useful tool in the framework of computational intelligence, since they provide a very interpretable model for the end user [1]. 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), containing the membership functions of the fuzzy partitions associated to the linguistic variables. 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 problem of imbalanced data-sets [2] 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 [3]. We must stress the importance of imbalanced data-sets, since such type of data appears in most of the real domains of classification such as medical diagnosis, finances, bioinformatics, image recognition and so on. The good behavior of FRBCS when dealing with imbalanced data-sets has been recently analyzed in [4].

Unfortunately, the use of FRBCSs in problems that presents a high number of features can originate RBs with a large 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 feature selection process that reduces the number of features used by the FRBCS.

On the other hand, 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 to choose the granularity level is not always appropriate since it 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 major 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 analyzed in [5] and some approaches for automatic learning of the KB in fuzzy modeling and fuzzy classification include the granularity learning [6], [7]. In a previous work [8], 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.

In [9], we proposed a Genetic Algorithm (GA) for jointly perform a feature selection and a granularity learning, considering a classical FRBCS learning method to derive the RB, the Chi et al.'s approach [10]. The main objective of that method was to obtain FRBCS for imbalanced data-sets with high prediction ability joint with a significative reduction of the model complexity in order to increase the FRBCS interpretability. To achieve that purpose, we considered two measures, one for the model accuracy and the other for the model complexity. We aggregated this two measures as the final fitness value of a standard generational GA (single objective). Therefore, the obtained models will have priority on high accuracy or low complexity depending on the relative weights of each measure in the final fitness function. As we were interested in the model with best prediction ability (highest value for the accuracy measure over the test data set), we tested various possibilities for these weighting factors, but the more adequate values are strongly dependent on the data-set considered. The results of that proposal were very promising but there were two drawbacks:

- Selecting the weighting factors. We proposed concrete values that obtained the best average in prediction ability of all the executions tested, but each data-set has its own optimal values.
- The GA provide only one solution. In some cases, designers are more interested in FRBCS with very low complexity rather than the highest prediction ability and it is necessary several executions of the GA with different weighting factors for obtaining the FRBCS desired.

It seems logical that these drawbacks can be avoided considering the two values of the fitness function as separate objectives in a Multiobjective Genetic Algorithm (MGA), that generate a set of FRBCSs with different trade-off between models with high accuracy (and normally, high complexity) and models with low complexity (and normally, low accuracy). The advantages of this approach are the elimination of parameters (weighting factors) and the extended set of different solutions obtained in one simple run.

In this paper we study the behavior of a multiobjective version of the GA presented in [9], using the well known Non dominated Sorting Genetic Algorithm (NSGA-II) [11]. In the analysis of results we will study the set of solutions obtained, with special regard to the relationship between the accuracy measure for the training data set and the accuracy measure for the test data set, in order to analyze the possible overfitting. This study can help us to deal with a new problem that appears with this multiobjective approach, the difficulty of determining the prediction ability (accuracy measure for the test data set) from the whole set of solutions obtained, as we will comment in the comparison of this approach with other methods that provide only one solution. There are some recent proposals that use a multiobjective GA for designing Fuzzy systems in classification and regression [12], [13], [14], [15], but these approaches only analyzes the prediction ability in one or a few solutions of the whole set of solutions provided.

We have selected a large collection of imbalanced data-sets from KEEL data-set repository[1] [16] for developing our experimental analysis. In order to deal with the problem of imbalanced data-sets we will make use of a preprocessing technique, the "Synthetic Minority Over-sampling Technique" (SMOTE) [17], to balance the distribution of training examples in both classes.

This paper is organized as follows. First, Section II and Section III introduce, respectively, the preliminary concepts of FRBCSs and imbalanced data-sets used in this paper. Next, in Section IV we will expose our proposal, a MGA for feature selection and granularity learning. Section V describes the experimental study and finally, in Section VI, some conclusions will be pointed out.

II. Fuzzy Rule Based Classification Systems

Any classification problem consists of m training patterns $x_p = (x_{p1}, \ldots, x_{pn})$, $p = 1, 2, \ldots, m$ 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:

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

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 [18]. 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 [4], [19], [8], [9]: the Chi *et al.*'s rule generation method [10]. This FRBCS design method is an extension of the well-known Wang and Mendel method [20] to classification problems.

III. Basic concepts on imbalanced data-sets

Learning from imbalanced data is an important topic that has recently appeared in the Machine Learning community [2]. We refer to imbalanced data when the class distribution is not uniform. In this situation, the number of examples that represents one of the classes of the data-set (usually the concept of interest) is much lower than that of the other classes.

Standard classifier algorithms have a bias towards the majority class, since the rules that predicts the higher number of examples are positively weighted during the learning process in favor of the accuracy metric. Consequently, the instances that belongs to the minority class are misclassified more often than those belonging to the majority class [21].

We will use the imbalance ratio (IR) [22] as a threshold to categorize the different imbalanced scenarios, which is defined as the ratio of the number of instances of the majority class and the minority class. We consider that a data-set presents a high degree of imbalance when its IR is higher than 9 (less than 10% of positive instances).

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

In a previous work on this topic [4], we analysed the cooperation of some preprocessing methods with FRBCSs, showing a good behavior for the oversampling methods, specially in the case of the SMOTE methodology [17]. We will employ in this contribution the same SMOTE algorithm in order to deal with imbalanced data-sets. In short, its main idea is to form new minority class examples by interpolating between several minority class examples that lie together. Thus, the overfitting problem is avoided and causes the decision boundaries for the minority class to spread further into the majority class space.

Most of proposals for automatic learning of classifiers use some kind of accuracy measure like the classification percentage over the example set. However, these measures can lead to erroneous conclusions working with imbalanced data-sets since it doesn't take into account the proportion of examples for each class. So, in this work we use the Area Under the Curve (AUC) metric [23], which can be defined as

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

where TP_{rate} is the percentage of positive cases correctly classified as belonging to the positive class and FP_{rate} is the percentage of negative cases misclassified as belonging to the positive class.

IV. Multiobjective Genetic Algorithm for Feature Selection and Granularity Learning

The proposed method is a multiobjective approach of a genetic process for the DB learning that allows us to select a set of variables (feature selection) and learn an adequate number of labels for each selected variable (granularity learning), that was proposed in [9]. The possible values considered for the granularity are taken from the set $\{2, \ldots, 7\}$. Once the granularity for each selected 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 a previous section that the RB learning algorithm used in this work is the method proposed in [10], that we have called the Chi et al.'s rule generation method.

We denote our proposal as MGA-FS-GL (Multiobjective GA for Feature Selection and Granularity Learning). The main purpose of MGA-FS-GL is to obtain a set of FRBCSs with different trade-off between good accuracy and reduced complexity taking the feature selection and granularity learning as a base. We have used the well known Non dominated Sorting Genetic Algorithm (NSGA-II [11]) as MGA scheme. Next, we describe the main components of MGA-FS-GL.

A. Encoding the DB

For a classification problem with *N* variables, 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:

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

It would be possible to merge both parts considering only an integer array, for example, including the value 1 as a placeholder for not using the variable. We use the two parts coding scheme to assign the same importance to both parts and to make easy the possibility of removing features as the mutation on the first part of the chromosome changes a selected variable for a non selected variable.

B. Initial Gene Pool

The initial population is composed of six groups with a different number of selected variables. Let g be 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. 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 than 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). We try to cover a wide zone of the search space with this population.

C. 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 two measure values of the resultant FRBCS:
  - Accuracy measure: AUC metric over the training data-set (AUC_{Tr}).
  - Complexity measure: the sum of the granularity levels of all the selected variables (denoted by Ng

in the following). By using this values as complexity measure, as an example, the following situations are considered 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

Finally, MGA-FS-GL considers two objectives to be minimized:

- $1 - AUC_{Tr}$ (as accuracy objective)
- Ng/N (as complexity objective)

D. Genetic operators

1) 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.

2) 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.

V. EXPERIMENTAL STUDY

We will study the performance of MGA-FS-GL employing a large collection of imbalanced data-sets with different imbalance ratio (IR). We divide the data-sets in two main groups:

- Data-sets with high imbalance :$IR > 9$
- Data-sets with low imbalance :$IR < 9$

Specifically, we have considered forty-four data-sets with different IR (twenty-two of each group) from KEEL data-set repository [16], which are publicly available on the corresponding web-page (http://www.keel.es/dataset.php), including general information about them. Table I show these data-sets, 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 according to the IR. For the sake of obtaining binary imbalanced problems, the positive and negative classes are defined as the joint of one or more classes, which are specified in column *Class* of Table I separated by a semicolon. In order to reduce the effect of imbalance, we will employ the SMOTE preprocessing method [17] for all our experiments, considering only the 1-nearest neighbor to generate the synthetic samples, and balancing both classes to the 50% distribution.

To develop the different experiments we consider a *5-folder cross-validation model*, i.e., 5 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. The data partitions used in this paper can be found in KEEL-dataset repository [41].

The configuration for the FRBCS is presented in Table II 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 [4]:

TABLE II
CONFIGURATION FOR THE FRBCS

Conjunction operator:	Product T-norm
Rule Weight:	Penalized Certainty Factor [18]
Fuzzy Reasoning Method:	Winning Rule

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

- Number of evaluations: $1000 \cdot N$
- Population Size: 100 individuals
- Crossover Probability P_c : 1.0
- Mutation Probability P_m : 0.2

The final result of MGA-FS-GL is a set of non dominated solutions i.e., none of them is better than another in the two objectives. This set is called the *Pareto front*. The usual way to analyze the results of a classification model in machine learning involves a comparison among the obtained model and the models obtained for other learning methods used in the specialized literature, including an statistical analysis of various measures being the prediction ability (in our case, the AUC_{Tst}) the main one. So, only one solution for each data-set is used in the comparison. In the case of a cross-validation scheme, it is used the mean of the unique solution obtained for each combination of training set/test set.

Therefore, it is difficult to show, in the previous way, the results of MGA-FS-GL because there are not five solutions for each data-set but five sets of non dominated solutions and the cardinality of that sets (Pareto fronts) is usually different. Thus, it is not possible to establish a comparison among the five sets of FRBCSs obtained for each data-set due to the different number of non dominated solutions and it is not possible to compare one solution (obtained for the reference models) opposite to a set of solutions. A possible way to deal with this problem was proposed in [12], where three different non dominated solutions were selected from the whole Pareto fronts and then, these models can be used as individual solutions for comparison with classical methods:

- The best solution for the accuracy objective (greatest value in AUC_{Tr}), named best accuracy (BA)

TABLE I
SUMMARY DESCRIPTION FOR IMBALANCED DATA-SETS

Data-set	#Ex.	#Atts.	Class (min., maj.)	%Class(min.; maj.)	IR
Data-sets with ow Imbalance (IR 1.5 to 9)					
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
Data-sets with High Imbalance (IR higher than 9)					
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

- The solution with the worst value for the accuracy objective (lowest value in AUC_{Tr}). So, the one with the less value for the complexity objective, named best interpretability (BI)
- The solution with the intermediate value for both objectives, named intermediate (I)

Therefore, these chosen solutions can be considered a brief sample of the whole set of solutions with different trade-off between accuracy and interpretability. Table III present the detailed results for these three selected solutions, showing the expected trend in the prediction ability: the most accurate solutions present certain overfitting. The intermediate solutions have a better behavior, since they present a good prediction ability (AUC_{Tst} lightly worse than the previous) and better interpretability (less number of rules) than the previous. The solutions with best interpretability have a very poor prediction ability since they are models with very few number of rules (normally, with only one feature selected).

One way to obtain more accurate solutions with good interpretability is to choose the FRBCS with at least two variables selected, and with the lowest value for the complexity measure. In fact, it would be easy to change MGA-FS-GL to ignore the models with only one selected feature, reducing the search space. However, we think that it is interesting to obtain these models as they provide information on what is the most relevant variable and, in some data-sets, the solution with best interpretability present a good prediction ability (*Yeast6, Yeast2vs4*) and the best prediction ability in one case (*Yeast05679vs4*).

Now, We will compare these solutions of MGA-FS-GL with the models obtained for the following methods:

- The method proposed in [9] (denoted GA-FS-GL). The single objective GA version of MGA-FS-GL where the fitness function is the weighted aggregation of the two objective values.

TABLE III
DETAILED TABLE OF RESULTS FOR THE SELECTED SOLUTIONS OF MGA-FS-GL

Dataset	Best-Accuracy			Intermediate			Best-Interpretability			#Non-Dom.
	Auc_{Tr}	Auc_{Tst}	#Rules	Auc_{Tr}	Auc_{Tst}	#Rules	Auc_{Tr}	Auc_{Tst}	#Rules	
Data-Sets with Low Imbalance (1.5 $<=$ IR $<$ 9)										
Glass1	83.14	62.85	79.20	72.86	65.93	33.20	49.75	48.31	1.60	10.40
Ecoli0vs1	98.70	95.66	62.80	84.40	86.58	12.40	66.34	66.22	2.00	6.20
Wisconsin	99.94	52.25	400.60	94.61	90.30	249.80	46.57	45.88	2.80	12.60
Pima	88.35	66.50	527.20	78.02	69.30	242.80	58.24	56.94	1.80	15.00
Iris0	99.63	99.50	9.20	95.38	93.50	5.20	88.00	88.00	2.00	4.80
Glass0	85.64	71.75	99.60	74.12	72.32	27.80	50.00	50.00	3.00	7.20
Yeast2	74.42	69.86	319.60	69.81	68.46	94.00	55.31	54.30	2.20	10.20
Vehicle2	98.12	86.64	798.60	86.21	81.62	397.60	49.86	49.77	1.80	12.20
Vehicle1	94.62	67.09	823.20	73.96	64.67	397.20	50.11	50.44	1.20	18.00
Vehicle3	94.82	60.77	829.80	73.04	65.25	448.60	50.00	50.00	1.60	17.80
Haberman	73.68	55.02	63.20	65.58	61.50	14.40	51.64	46.75	2.20	11.00
Glass0123vs456	99.29	78.67	105.20	89.35	87.42	29.00	47.24	47.23	4.20	11.40
Vehicle0	98.47	84.23	818.00	87.59	85.60	306.80	50.00	50.00	6.00	16.40
Ecoli1	94.31	84.88	132.80	80.60	78.57	31.00	75.70	75.57	2.00	10.00
New-Thyroid2	99.31	96.63	39.60	91.01	92.66	12.40	76.56	77.14	2.00	7.20
New-Thyroid1	99.93	93.45	47.00	92.50	88.06	11.40	76.55	76.83	2.00	7.60
Ecoli2	95.14	84.77	155.60	87.69	84.74	32.20	49.84	46.73	4.40	10.40
Segment0	99.41	97.62	801.60	88.31	87.90	496.60	52.06	51.73	3.00	12.60
Glass6	98.59	84.23	107.20	94.45	86.80	53.80	64.88	66.22	2.00	9.80
Yeast3	94.79	91.41	334.80	83.41	83.40	79.80	61.37	60.94	2.00	8.20
Ecoli3	95.93	89.65	154.80	88.87	89.62	21.40	79.23	79.03	2.00	9.00
Page-Blocks0	89.74	87.27	292.20	81.17	80.04	48.80	49.93	49.82	2.40	12.00
Mean	93.45	80.03	318.26	83.31	80.19	138.46	59.05	58.54	2.46	10.91
Data-Sets with High Imbalance (IR $>=$ 9)										
Yeast2vs4	95.20	86.86	205.80	90.34	89.04	25.40	82.32	81.96	2.00	8.80
Yeast05679vs4	90.96	73.72	248.40	83.42	76.17	95.00	79.60	79.54	2.00	11.60
Vowel0	99.92	95.55	698.40	95.42	93.99	278.80	49.99	50.00	1.20	17.20
Glass016vs2	83.73	58.52	95.00	70.91	60.64	31.60	49.93	50.00	2.00	12.80
Glass2	83.63	55.83	96.00	60.61	55.96	51.40	49.94	50.00	2.20	10.60
Ecoli4	98.62	89.81	110.20	85.86	78.88	15.80	49.92	49.69	4.00	11.00
shuttle0vs4	99.98	99.10	32.00	99.66	99.11	7.80	50.00	49.58	1.40	5.60
yeastB1vs7	86.98	67.21	210.60	72.20	71.48	34.60	61.79	60.61	2.00	12.00
Glass4	99.38	86.43	114.00	82.27	78.77	38.20	16.92	16.93	3.00	11.20
Page-Blocks13vs4	99.30	94.30	133.60	79.86	80.49	95.20	50.00	50.00	2.00	8.80
Abalone9-18	79.59	73.24	131.40	55.79	54.91	3.20	50.00	50.00	2.80	8.40
Glass016vs5	98.57	83.43	120.00	94.57	87.57	77.40	50.00	50.00	3.20	14.20
shuttle2vs4	100.00	89.18	24.60	81.35	76.80	5.00	49.59	50.00	4.00	7.80
Yeast1458vs7	86.14	64.84	284.80	71.97	64.41	60.00	60.98	61.74	2.00	12.60
Glass5	98.90	73.90	100.00	88.61	77.20	27.40	56.95	57.56	4.40	10.40
Yeast2vs8	89.00	70.55	136.20	74.52	73.52	24.60	50.39	44.82	2.20	9.60
Yeast4	90.14	85.15	318.40	83.41	81.15	53.00	49.94	49.86	4.00	10.40
Yeast1289vs7	84.99	74.45	233.80	66.05	64.96	42.60	59.55	60.56	2.00	11.00
Yeast5	96.94	92.81	281.80	90.13	89.41	56.40	51.19	50.52	3.00	6.80
Yeast6	93.05	88.06	229.40	87.92	88.04	16.60	86.76	86.83	2.00	6.40
Ecoli0137vs26	97.72	70.45	148.60	90.53	81.54	23.80	50.05	50.00	3.00	10.00
Abalone19	80.09	62.56	267.20	69.92	67.93	63.00	55.19	55.09	2.00	9.20
Mean	92.40	78.91	191.83	80.70	76.91	51.22	55.04	54.79	2.56	10.29
All Data-Sets										
Global	92.93	79.47	255.05	82.00	78.55	94.84	57.05	56.66	2.51	10.60

- The original Chi et al.'s method [10], where a previous definition for the DB is needed, normally by the use of 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, G7-Chi, respectively.
- The method proposed in [8] (denoted GA-GL), that uses a single objective GA (similar to the used in GA-FS-GL) only for granularity learning.
- C4.5 [24], a method of reference in the field of classification with imbalanced data-sets [25], [22].

All the previous methods have all the variables selected except GA-FS-GL, that includes a feature selection in the same way as MGA-FS-GL. The FRBCS approaches (GA-FS-GL, GA-GL, G3-Chi, G5-Chi and G7-Chi) have the same configuration as MGA-FS-GL (Table II)

Table IV shows the mean of the AUC and the number of rules (NR) for the different groups of data-sets (Low imbalance, High Imbalance and all of them). There are three lines for MGA-FS-GL, corresponding with the three solutions selected: best accuracy (BA), intermediate (I) and best interpretability (BI).

As can be observed in Table IV for all the data-sets, the single objective version of the genetic learning process described (GA-FS-GL) presents the best results. Its prediction ability is better than the other methods except for GA-GL. A GA designed only for granularity learning (GA-GL) obtains the best results in prediction ability but with a high increase of the number of rules (less interpretability). Regarding to the complexity of the models (lower NR), GA-FS-GL presents

TABLE IV

AVERAGE TABLE OF RESULTS IN AUC_{Tr}, AUC_{Tst} AND NUMBER OF RULES (NR) FOR THE DIFFERENT DEGREES OF IMBALANCE

Algorithm	Low Imbalance			High Imbalance			All Data-sets		
	AUC_{Tr}	AUC_{Tst}	NR	AUC_{Tr}	AUC_{Tst}	NR	AUC_{Tr}	AUC_{Tst}	NR
G3-Chi	86.03	82.14	124.70	85.09	80.52	68.67	85.56	81.33	96.69
G5-Chi	91.60	81.43	276.93	89.56	78.76	160.20	90.58	80.10	218.56
G7-Chi	93.95	74.40	387.18	93.66	74.90	272.02	93.81	74.65	329.60
GA-GL	92.03	84.20	177.41	91.33	81.98	82.36	91.68	83.09	129.89
GA-FS+GL	85.77	83.53	26.98	85.33	80.91	12.79	85.55	82.22	19.89
C4.5	95.00	86.09	26.05	95.93	78.25	22.45	95.46	82.17	24.25
MGA-FS+GL(BA)	93.45	80.03	318.26	92.40	78.91	191.83	92.93	79.47	255.05
MGA-FS+GL(I)	83.31	80.19	138.46	80.70	76.91	51.22	82.00	78.55	94.84
MGA-FS+GL(BI)	59.05	58.54	2.46	55.04	54.79	2.56	57.05	56.66	2.51

the lesser number of rules, except for the solution with best interpretability of MGA-FS-GL, that presents the worse value for the prediction ability. The improvement of GA-FS-GL in the number of rules is even better in the group of data-sets with high imbalance, demonstrating its suitability to the framework of imbalanced data-sets.

Anyway, None of the three selected solutions of MGA-FS-GL obtain better results on AUC_{Tst} than the obtained with GA-FS-GL (considering the weighting factors proposed in [9]). However, MGA-FS-GL provide models with very good prediction ability in each Pareto front. If we chose the solutions with best AUC_{Tst} for each set of non dominated solutions, the global mean of the AUC_{Tst} for all the data-sets would be 86.32, significantly higher than those obtained with GA-FS-GL and the other methods used for comparison. Hence, MGA-FS-GL performs very well, but the unsolved problem is to find these solutions with the best prediction ability as they occupy different positions along the Pareto front of each data-set. This situation is represented in Figure 1 where the Pareto fronts obtained for MGA-FS-GL in the first partition of four problems are showed (including also the AUC_{Tst} of each non dominated solution). There are different "types" of behavior among the data-sets regarding to the values of AUC_{Tst}:

- In some cases, the values of AUC_{Tst} follow the same trend as AUC_{Tr} (Vowel0). In such data-sets, the trade-off between accuracy and interpretability is also fulfilled for the AUC_{Tst} and the final user can choose the most suitable FRBCS (depending on his preferences) assuming that AUC_{Tr} is similar to the prediction ability in all models. Unfortunately, this situation occurs only in a few data-sets.
- Many data-sets present overfitting, the AUC_{Tst} is then increased from the FRBCS with best AUC_{Tr} to an optimal value from which descends again. This point corresponds to a particular combination of weights in GA-FS-GL, which is specific for each data-set. Therefore, finding this value is a difficult problem to solve. In some cases there are a great overfitting in the solutions with best AUC_{Tr} (Wisconsin) while in other cases the overfitting is lesser and there is a smooth decrease of the AUC_{Tst} from the optimal value (Ecoli3).
- Finally, in other cases, the AUC_{Tst} follows a chaotic behavior (Glass4). So, the data fracture between the training set and the test set is very high and it is impossible to determine the solution with best prediction ability.

As most of the data-sets belong to one of the last two "types", we recommend using GA-FS-GL when the main purpose is to obtain only one model with high prediction ability (and not much complexity), unless there are some evidence to establish that the problem belongs to the first category of the previously described. MGA-FS-GL is particulary useful when many solutions with different trade-off between accuracy and interpretability is required, but it is very difficult to find the solution with best prediction ability, as that measure can not be used in the learning process. Moreover, it is difficult to know the behavior of the accuracy measure in the test data set (compared with the known behavior in training data set), that it would be valuable for selecting the most adequate solution if we were interested, for example, in a solution with a minimum degree of interpretability.

VI. CONCLUSIONS

This contribution analyzes the behavior of a multiobjective version of a previously proposed Genetic Algorithm to design FRBCS for imbalanced data-sets. The genetic learning process 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.

The proposed approach obtains a set of FRBCSs with different trade-off between accuracy and interpretability in which to find the model with the best prediction ability generally is non evident.

ACKNOWLEDGMENT

This work has been supported by the Spanish ministry of science and technology under project TIN2008-06681-C06-01

REFERENCES

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

[2] 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.

[3] 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.

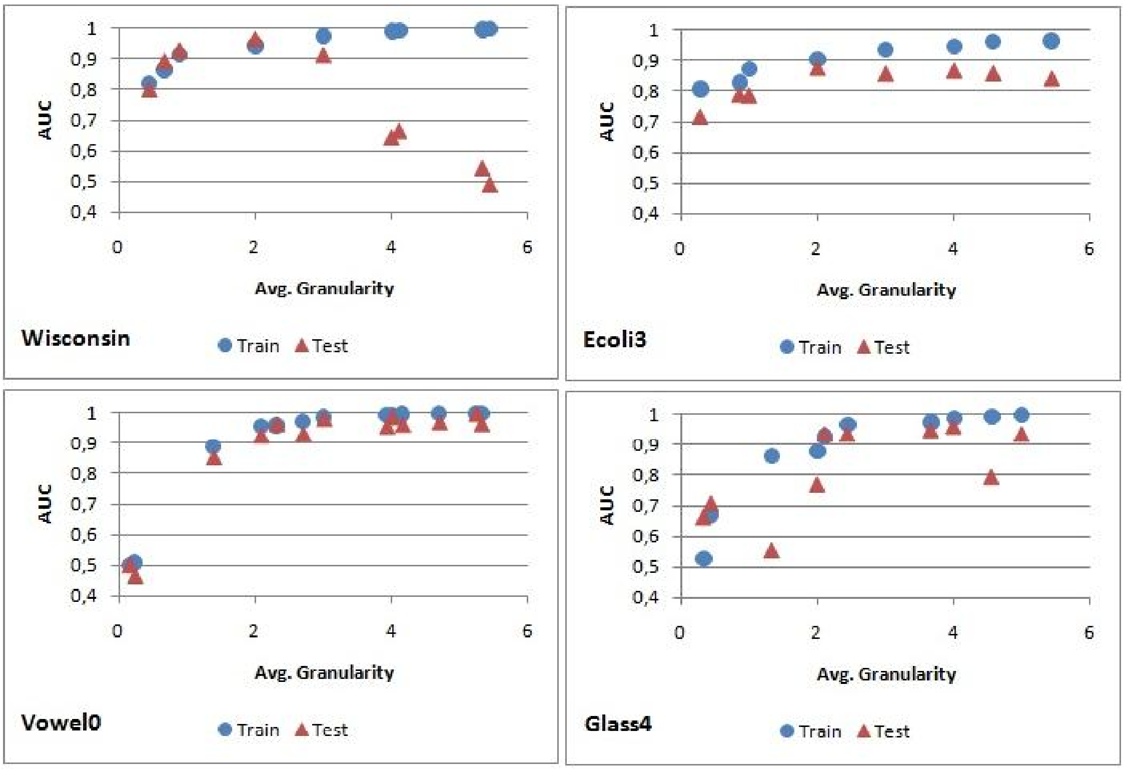

Fig. 1. Pareto fronts

[4] 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.

[5] 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.

[6] ——, "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.

[7] 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.

[8] 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 (F ZZ–IEEE09)*, 2009, pp. 1689–1694.

[9] ——, "A genetic algorithm for feature selection and granularity learning in fuzzy rule-based classification systems for highly imbalanced data-sets," in *13th International Conference on Information Processing and Management of ncertainty in Knowledge-Based Systems (IPM 2010)*, 2010, pp. 741–750.

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

[11] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA–II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[12] R. Alcalá, P. Ducange, F. Herrera, B. Lazzerini, and F. Marcelloni., "A multiobjective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy rule-based systems," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 5, pp. 1106–1122, 2009.

[13] F. M. M. Antonelli, P. Ducange and B. Lazzerini, "Learning concurrently partition granularities and rule bases of mamdani fuzzy systems in a multi-objective evolutionary framework," *International Journal of Approximate Reasoning*, vol. 50, no. 7, pp. 1066–1080, 2009.

[14] 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.

[15] P. Pulkkinen and H. Koivisto, "A dynamically constrained multiobjective genetic fuzzy system for regression problems," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 161–177, 2010.

[16] 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- alued ogic and Soft Computing*, vol. in press, 2011.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] G. M. Weiss, "Mining with rarity: a unifying framework," *SIGKDD Explorations*, vol. 6, no. 1, pp. 7–19, 2004.

[22] A. Orriols-Puig and E. Bernadó-Mansilla, "Evolutionary rule–based systems for imbalanced datasets," *Soft Computing*, vol. 13, no. 3, pp. 213–225, 2009.

[23] 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.

[24] J. R. Quinlan, *C4.5: Programs for Machine earning*. San Mateo–California: Morgan Kaufmann Publishers, 1993.

[25] 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.