A Genetic Algorithm for Feature Selection and Granularity Learning in Fuzzy Rule-Based Classification Systems for Highly Imbalanced Data-Sets

Pedro Villar[1], Alberto Fernández[2], and Francisco Herrera[2]

[1] Department of Software Engineering,
[2] Department of Computer Science and Artificial Intelligence,
E.T.S. Ing. Informática y de Telecomunicación, University of Granada
pvillarc@ugr.es, alberto@decsai.ugr.es, herrera@decsai.ugr.es

Abstract. This contribution proposes a Genetic Algorithm for jointly performing a feature selection and granularity learning for Fuzzy Rule-Based Classification Systems in the scenario of data-sets with a high imbalance degree. 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 and precise models by selecting the adequate variables and adapting the number of fuzzy labels for each problem.

Keywords: Fuzzy Rule-Based Classification Systems, imbalanced data-sets, Genetic Algorithms, feature selection, granularity level.

1 Introduction

The problem of imbalanced data-sets [1] for binary classification occurs when the number of instances for each class are very different between them, and usually the less representative class is the one which has more interest from the point of view of the learning task. We develop an experimental analysis in the context of imbalance classification for binary data-sets when the class imbalance ratio is high. 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 [2]. The good behavior of FRBCS when dealing with imbalanced data-sets has been recently analysed in [3].

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.

E. Hüllermeier, R. Kruse, and F. Hoffmann (Eds.): IPMU 2010, Part I, CCIS 80, pp. 741–750, 2010.

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 tackled from a double perspective: a reduction of the rule set, minimising the number of fuzzy rules included in the RB or a feature selection process that reduces the number of features used by the FRBCS. 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. It increases the efficiency and accuracy of the learning and classification stages.

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. However, 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. In a previous work [4], we analyse the influence of granularity learning in the performance of FRBCSs for imbalanced data sets, and the results obtained show that is possible an significant improvement in the classification ability only by learning an adequate number of labels per variable although the complexity of the model was lightly increased.

Our objective is to propose a genetic learning process to improve the prediction ability of the FRBCSs for imbalanced data-sets joint with a significative reduction of the model complexity in order to increase the FRBCS interpretability. Our proposal uses a Genetic Algorithm (GA) for jointly perform a feature selection and a granularity learning, and considers a classical FRBCS learning method to derive the rule base, the Chi et al.'s approach [5]. 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 an adequate granularity level. We also want to check the performance of our method compared with a non-FRBCS classification model, C4.5 [6], a decision tree algorithm that has been used as a reference in the imbalanced data-sets field [7].

We have selected a large collection of data-sets with high imbalance from UCI repository [8] 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) [9], to balance the distribution of training examples in both classes. Furthermore, we will perform a statistical study using non-parametric tests [10] to find significant differences among the obtained results.

This contribution is organized as follows. First, Section 2 introduces the problem of imbalanced data-sets, describing its features, how to deal with this problem and the metric we have employed in this context. Next, in Section 3 we will expose the characteristics of our proposal, a GA for feature selection and granularity learning. Section 4 contains the experimental study. Finally, in Section 5, some conclusions will be pointed out.

2 Imbalanced Data-Sets in Classification

Learning from imbalanced data is an important topic that has recently appeared in the Data Mining community [1]. This problem is very representative since it appears in a variety of real-world applications including, but not limited to, medical applications, finance, telecommunications, biology and so on. 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. We will use the imbalance ratio (IR) [11] 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).

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 favour of the accuracy metric. Consequently, the instances that belongs to the minority class are misclassified more often than those belonging to the majority class [12].

In a previous work on this topic [3], we analysed the cooperation of some preprocessing methods with FRBCSs, showing a good behaviour for the oversampling methods, specially in the case of the SMOTE methodology [9]. According to this, we will employ in this contribution the 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.

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. Therefore, in this work we use the Area Under the Curve (AUC) metric [13], which can be defined as $(1 + TP_{rate} - FP_{rate})/2$, 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.

3 Genetic Algorithm for the Data Base Learning

In this section, we propose an standard generational GA for the DB that allows us to select a set of variables (feature selection) and learn an adequate number

of labels for each selected variable (granularity learning). Once the granularity for each selected feature are 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 chromosome can be evaluated. The RB derivation algorithm used in this work is the method proposed in [5], that we have called the Chi et al.'s method.

We denote our proposal as GA-FS-GL (Genetic Algorithm for Feature Selection and Granularity Learning). 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 this Section. Next, we describe the main components of GA-FS-GL.

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_1): 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_2): the number of labels per variable is stored in an integer array of length N. In this contribution, the possible values considered 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:

$$C_1 = (v_1, v_2, \ldots, v_N) \quad C_2 = (g_1, g_2, \ldots, g_N) \qquad C = C_1 C_2$$

Initial Gene Pool. The initial population is composed of six groups with a different number of selected variables. Next, we describe its generation:

- In the first group all the chromosomes have all the features selected. It is composed of two parts. In the first part all the chromosomes have the same granularity in all its variables and it is composed of g chromosomes, with g being the cardinality of the significant term set, in our case $g = 6$, corresponding to the six possibilities for the number of labels, $2 \ldots 7$. For each granularity level, one individual is created. The second part is composed of 10 chromosomes and 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: $(g + 10) \times 5$.

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: The usual way to proceed in this type of genetic learning is to choose a kind of accuracy measure over the training data-set, like the AUC metric. However, as mentioned before, we will lightly penalize FRBCSs with high number of selected variables and/or high granularity levels in order to avoid the possible overfitting, thus improving the generalization capability of the final FRBCS. To do that, once the RB has been generated and its AUC over the training set has been calculated, the fitness function to be minimized is:

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

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:

$$\omega_2 = \alpha_{\omega_2} \cdot \frac{AUC_{max\_g}}{max\_g}$$

with α_{ω_2} being a weighting percentage.

Genetic Operators

– **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.
– **Crossover:** the crossover works in the two parts of the chromosome at the same time. Therefore, an standard crossover operator is applied over C_1 and C_2. This operator performs as follows: a crossover point p is randomly generated in C_1 and the two parents are crossed at the p-th variable in C_1 (the possible values for p are $\{2, \ldots, N\}$). The crossover is developed this way in the two chromosome parts, C_1 and C_2, thereby producing two meaningful descendants.

– **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_1: As this part of the chromosome is binary coded, a simple binary mutation is developed, flipping the value of the gene.
  - *Mutation on* C_2: The mutation operator selected for C_2 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

We will study the performance of GA-FS-GL employing a large collection of imbalanced data-sets with a high imbalance ratio ($IR > 9$). Specifically, we have considered twenty-two data-sets from UCI repository [8] with different IR, as shown in Table 1, 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. 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 [9] for all our experiments, considering only the 1-nearest neighbour to generate the synthetic samples, and balancing both classes to the 50% distribution.

We will analyse the influence of feature selection and granularity learning by means of a comparison between the performance of GA-FS-GL and two FRBCS models obtained by Chi et al.'s method with all the variables selected:

– The original Chi et al.'s method, 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. So, it is necessary to choose a number of labels. The usual values employed for Chi et al.'s approach in the specialized literature are 3 and 5 labels per variable. Previous experiments [4] showed that the FRBCSs with three labels for variable obtain better results in prediction ability (less value in AUC for the test data set) and interpretability (less number of rules) so we choose this granularity level for the comparison. In the latter, we will refer that method as G3-Chi.
– The method proposed in [4] (denoted GA-GL), that uses a GA for granularity learning and the Chi et al.'s method to derive the RB.

As mentioned before, we also compare the results of GA-FS-GL with C4.5 [6], a method of reference in the field of classification with imbalanced data-sets [7]. The configuration for the FRBCSs approaches, GA-FS-GL, GA-GL and Chi et al.'s, is presented below. 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 [3]:

Table 1. 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

- Conjunction operator to compute the compatibility degree of the example with the antecedent of the rule: Product T-norm.
- Rule Weight: Penalized Certainty Factor [14].
- Conjunction operator between the compatibility degree and the rule weight: Product T-norm.
- Fuzzy Reasoning Method: Winning Rule.

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. 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. Furthermore, Wilcoxon's Signed-Ranks Test [15] is used for statistical comparison of our experimental results. 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
-Parameters of the evaluation function (Section 3): (ω_1 : 0.7 , α_{ω_2} : 0.3)

Table 2 shows the results in performance (using the AUC metric) for GA-FS-GL and the algorithms employed for comparison, that is, G3-Chi, GA-GL and C4.5, being AUC_{Tr} the AUC over the training data-set and AUC_{Tst} the AUC over the test data-set. The final line of the table shows the mean of the number of rules (NR) of the classifiers.

Table 2. Detailed results table for the problems considered

Data-set	G3-Chi		GA-GL		GA-FS-GL		C4.5	
	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
Yeast2vs4	89.68	87.36	93.79	90.84	94.38	94.52	98.14	85.88
Yeast05679vs4	82.65	79.17	86.11	81.78	83.37	78.97	95.26	76.02
Vowel0	98.57	98.39	99.59	99.07	96.58	96.49	99.67	94.94
Glass016vs2	62.71	54.17	85.96	60.54	78.23	56.07	97.16	60.62
Glass2	66.54	55.30	83.71	57.42	79.42	56.88	95.71	54.24
Ecoli4	94.06	91.51	98.14	90.90	93.20	92.31	97.69	83.10
Yeast1vs7	82.00	80.63	82.43	75.79	77.37	70.75	93.51	70.03
Shuttle0vs4	100.00	99.12	100.00	99.42	100.00	99.97	99.99	99.97
Glass4	95.27	85.70	98.71	87.92	95.02	85.20	98.44	85.08
Page-Blocks13vs4	93.68	92.05	99.59	99.10	98.25	96.99	99.75	99.55
Abalone9vs18	70.23	64.70	82.38	73.68	78.63	68.18	95.31	62.15
Glass016vs5	90.57	79.71	98.21	85.43	95.50	84.57	99.21	81.29
Shuttle2vs4	95.00	90.78	99.73	94.25	99.09	98.78	99.90	99.17
Yeast1458vs7	71.25	64.65	85.69	65.47	76.00	74.67	91.58	53.67
Glass5	94.33	83.17	98.03	79.92	94.57	79.15	99.76	88.29
Yeast2vs8	78.61	77.28	84.57	79.32	81.69	79.46	91.25	80.66
Yeast4	83.58	83.15	86.90	80.66	84.47	80.31	91.01	70.04
Yeast1289vs7	74.70	77.12	80.27	70.98	76.00	74.67	94.65	68.32
Yeast5	94.68	93.58	96.48	94.73	95.58	93.54	97.77	92.33
Ecoli0137vs26	93.96	81.90	97.69	81.36	97.22	80.99	96.78	81.36
Yeast6	88.48	88.09	91.09	86.06	89.37	87.01	92.42	82.80
Abalone19	71.44	63.94	80.28	69.03	77.40	73.16	85.44	52.02
Mean	85.09	80.52	91.33	81.98	88.39	81.42	95.93	78.25
NR mean	68.67		82.36		37.31		22.45	

As it can be observed, the performance obtained by GA-FS-GL is higher than the one for G3-Chi, both in AUC_{Tr} and AUC_{Tst}, showing the significative influence of the feature selection and granularity level in the behaviour of the classifier. GA-FS-GL obtain results very similar to GA-GL in AUC (Table 3 shows no significative differences between them in AUC_{Tst}) but the number of rules is very much lower in GA-FS-GL by the feature selection process, reducing the complexity of the model. Therefore, the interpretability of the FRBCSs generated by GA-FS-GL is greater than the other methods. Furthermore, GA-FS-GL present better results than C4.5 in AUC_{Tst}. This situation is represented statistically by means of a Wilcoxon test (Table 3, with R^+ corresponds to GA-FS-GL and R^- to the other method).

Table 3. Wilcoxon test to compare the methods according to their performance

Comparison	R^+	R^-	p-value
GA-FS-GL vs. G3-Chi	150.5	102.5	0.436
GA-FS-GL vs. GA-GL	95.0	158.0	0.306
GA-FS-GL vs. C4.5	198.5	54.5	0.019

GA-FS-GL obtain 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 4 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

Table 4. Mean of number of selected variables and labels learned by GA-FS-GL

	Variables										
Data-set	SV	1	2	3	4	5	6	7	8	9	10
Yeast2vs4	2.0	1.0/3.0	.00/0.0	1.0/4.0	.00/0.0	.00/0.0	.00/0.0	.00/0.0	.00/0.0	-	-
Yeast05679vs4	2.8	1.0/2.4	.40/2.5	.60/2.0	.00/0.0	.80/2.0	.00/0.0	.00/0.0	.00/0.0	-	-
Glass016vs2	2.6	.40/4.5	.20/3.0	.00/0.0	.60/5.0	.60/5.7	.40/4.0	.20/7.0	.00/0.0	.20/3.0	-
Glass2	2.6	.40/5.0	.00/0.0	.40/2.0	.40/4.5	1.0/4.6	.00/0.0	.00/0.0	.00/0.0	.40/6.5	-
Ecoli4	2.0	.00/0.0	.60/2.0	.00/0.0	.00/0.0	1.0/3.0	.20/3.0	.20/2.0	-	-	-
Shuttle0vs4	2.0	.20/3.0	.20/4.0	.20/3.0	.00/0.0	.00/0.0	.20/3.0	.80/3.8	.40/4.5	.00/0.0	-
Yeast1vs7	2.2	.60/2.7	.00/0.0	1.0/2.6	.00/0.0	.00/0.0	.00/0.0	.20/2.0	.40/3.5	-	-
Glass4	2.4	.00/0.0	.00/0.0	.40/4.0	.60/3.7	.20/2.0	.00/0.0	.60/3.0	.60/3.3	.00/0.0	-
Pageblocks13vs4	2.0	1.0/4.4	.00/0.0	.00/0.0	.00/0.0	1.0/4.4	.00/0.0	.00/0.0	.00/0.0	.00/0.0	.00/0.0
Abalone9vs18	2.2	.40/2.0	.00/0.0	.00/0.0	.20/2.0	.00/0.0	.60/6.7	.00/0.0	1.0/5.8	-	-
Glass016vs5	2.8	.20/6.0	.40/3.0	1.0/3.6	.20/2.0	.00/0.0	.00/0.0	.20/3.0	.60/3.3	.20/3.0	-
Shuttle2vs4	2.8	.60/3.0	.00/0.0	1.0/3.0	.00/0.0	.00/0.0	.00/0.0	1.0/2.2	.00/0.0	.20/3.0	-
Yeast1458vs7	4.0	.60/5.3	.80/5.0	1.0/4.6	.60/5.3	.00/0.0	.00/0.0	.00/0.0	1.0/3.2	-	-
Glass6	2.4	.00/0.0	.40/3.5	1.0/3.2	.00/0.0	.00/0.0	.00/0.0	.20/3.0	.60/3.0	.20/4.0	-
Yeast2vs8	2.2	.80/4.0	.40/2.0	.00/0.0	.00/0.0	.00/0.0	1.0/2.0	.00/0.0	.00/0.0	-	-
Yeast4	2.6	1.0/3.0	.40/2.0	.80/3.0	.00/0.0	.40/2.0	.00/0.0	.00/0.0	.00/0.0	-	-
Yeast1289vs7	3.2	1.0/2.2	.00/0.0	1.0/3.2	.00/0.0	.00/0.0	.00/0.0	.20/5.0	1.0/2.2	-	-
Yeast5	2.8	1.0/3.2	.80/2.3	.60/2.0	.20/2.0	.20/2.0	.00/0.0	.00/0.0	.00/0.0	-	-
Yeast6	2.8	.80/3.0	.80/2.3	.00/0.0	.00/0.0	.40/2.0	.00/0.0	.20/2.0	.60/2.7	-	-
Ecoli0137vs26	3.2	1.0/3.6	.40/3.5	1.0/2.8	.00/0.0	.00/0.0	.40/4.5	.40/5.0	-	-	-
Abalone19	2.0	.00/0.0	.00/0.0	.20/3.0	.00/0.0	.00/0.0	.60/6.7	.20/3.0	1.0/5.8	-	-
Vowel0	2.2	.00/0.0 11 .00/0.0	.00/0.0 12 .00/0.0	.00/0.0 13 .00/0.0	.80/4.0	1.0/7.0	.20/3.0	.00/0.0	.20/2.0	.00/0.0	.00/0.0

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 4, 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. Moreover, in nineteen problems, less than three variables are selected in the average of the 15 executions. 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 contribution.

5 Conclusions

This contribution has proposed a method to design FRBCS with good accuracy and interpretability for imbalanced data-sets with a high imbalance ratio. 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. We must remark one advantage of our proposal, the GA can be combined with any rule generation method. We have used a simple algorithm for efficiency but another more accurate one can be used. Our future work will be focused on applying a multi-objective genetic algorithm in order to obtain a set of solutions with different trade-off between accuracy (high AUC) and

interpretability (low number of rules), eliminating the problem of the choice of weights in the fitness function.

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

References

1. Chawla, N.V., Japkowicz, N., Kolcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations 6(1), 1–6 (2004)
2. Ishibuchi, H., Nakashima, T., Nii, M.: Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining. Springer, Heidelberg (2004)
3. Fernández, A., García, S., Del Jesus, M.J., Herrera, F.: A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. Fuzzy Sets and Systems 159(18), 2378–2398 (2008)
4. Villar, P., Fernández, A., Herrera, F.: 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-IEEE 2009), pp. 1689–1694 (2009)
5. Chi, Z., Yan, H., Pham, T.: Fuzzy algorithms with applications to image processing and pattern recognition. World Scientific, Singapore (1996)
6. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993)
7. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A Study of the Behaviour of Several Methods for Balancing Machine Learning Training Data. SIGKDD Explorations 6(1), 20–29 (2004)
8. Asuncion, A., Newman, D.J.: UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
9. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligent Research 16, 321–357 (2002)
10. García, S., Herrera, F.: An Extension on "Statistical Comparisons of Classifiers over Multiple data sets" for all Pairwise Comparisons. Journal of Machine Learning Research 9, 2607–2624 (2008)
11. Orriols-Puig, A., Bernadó-Mansilla, E.: Evolutionary rule-based systems for imbalanced datasets. Soft Computing 13(3), 213–225 (2009)
12. Weiss, G.M.: Mining with rarity: a unifying framework. SIGKDD Explorations 6(1), 7–19 (2004)
13. Huang, J., Ling, C.X.: Using AUC and accuracy in evaluating learning algorithms. IEEE Transactions on Knowledge and Data Engineering 17(3), 299–310 (2005)
14. Ishibuchi, H., Yamamoto, T.: Rule Weight Specification in Fuzzy Rule-Based Classification Systems. IEEE Transactions on Fuzzy Systems 13, 428–435 (2005)
15. Sheskin, D.: Handbook of parametric and nonparametric statistical procedures, 2nd edn. Chapman & Hall/CRC, Boca Raton (2006)