Genetic Cooperative-Competitive Fuzzy Rule Based Learning Method using Genetic Programming for Highly Imbalanced Data-Sets

Alberto Fernández[1] Francisco J. Berlanga[2] María J. del Jesus[3] Francisco Herrera[1]

1.Department of Computer Science and Artificial Intelligence, University of Granada
Granada, Spain
2.Department of Computer Science and Systems Engineering, University of Zaragoza
Zaragoza, Spain
3.Department of Computer Science, University of Jaén
Jaén, Spain
Email: alberto@decsai.ugr.es, berlanga@unizar.es, mjjesus@ujaen.es, herrera@decsai.ugr.es

Abstract— *Classification in imbalanced domains is an important problem in Data Mining. We refer to imbalanced classification when data presents many examples from one class and few from the other class, and the less representative class is the one which has more interest from the point of view of the learning task. The aim of this work is to study the behaviour of the GP-COACH algorithm in the scenario of data-sets with high imbalance, analysing both the performance and the interpretability of the obtained fuzzy models. To develop the experimental study we will compare this approach with a well-known fuzzy rule learning algorithm, the Chi et al.'s method, and an algorithm of reference in the field of imbalanced data-sets, the C4.5 decision tree.*

Keywords— Fuzzy Rule-Based Classification Systems, Genetic Fuzzy Systems, Genetic Programming, Imbalanced Data-Sets, Interpretability

1 Introduction

In the area of Data Mining, real world classification problems present some features that can diminish the accuracy of Machine Learning algorithms, such as the presence of noise or missing values, or the imbalanced distribution of classes.

Specifically, the problem of imbalanced data-sets has been considered as one of the emergent challenges in Data Mining [1]. This situation occurs when one class is represented by a large number of examples (known as negative class), whereas the other is represented by only a few (positive class).

Our objective is to develop an empirical 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 Fuzzy Rule Based Classification Systems (FRBCSs), a very useful tool in the ambit of Machine Learning, since they provide a very interpretable model for the end user [2].

We will employ a novel approach, GP-COACH (Genetic Programming-based evolutionary algorithm for the learning of COmpact and ACcurate FRBCS) [3], that learns disjunctive normal form (DNF) fuzzy rules (generated by means of a context-free grammar) and obtains very interpretable FRBCSs, with few rules and conditions per rule, with a high-generalization capability.

We want to analyse whether this model is accurate for data-sets with high imbalance in contrast with an FRBCS, the Chi et al.'s approach [4] and with C4.5 [5], a decision tree algorithm that has been used as a reference in the imbalanced data-sets field [6, 7]. We will also focus on the tradeoff between accuracy and interpretability [8] for the final obtained models. We will employ the Area Under the Curve (AUC) metric [9] to compute the classification performance, whereas we will measure the interpretability of the system by means of the number of rules in the system.

We have selected a large collection of data-sets with high imbalance from UCI repository [10] for developing our empirical 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) [11], to balance the distribution of training examples in both classes. In this manner, we will analyse the positive synergy between the GP-COACH model and the SMOTE preprocessing technique for dealing with imbalanced data-sets. Furthermore, we will perform a statistical study using non-parametric tests [12, 13, 14] 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 present the GP-COACH algorithm, explaining in detail the characteristics of this novel approach. Section 4 contains the experimental study for GP-COACH, Chi et al.'s and C4.5 algorithms regarding performance and interpretability. Finally, Section 5 summarizes and concludes the work.

2 Imbalanced Data-Sets in Classification

Learning from imbalanced data is an important topic that has recently appeared in the Machine Learning community [15, 16, 17]. The significance of this problem consists in its presence in most of the real domains of classification, such as fraud detection [18], risk management [19] and medical applications [20] among others.

This problem occurs when the number of instances of one class is much lower than the instances of the other classes. In this situation, the class of interest is often the one with the smaller number of examples, whereas the other class(es) represent(s) the counterpart of that concept and, in that manner, include(s) a high amount of data.

Standard classifier algorithms have a bias towards the majority class, since the rules that predicts the larger number of

ISBN: 978-989-95079-6-8

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 [21]. Other important issue of this type of problem is the small disjuncts that can be found in the data-set [22] and the difficulty of most learning algorithms to detect those regions. Furthermore, the main handicap on imbalanced data-sets is the overlapping between the examples of the positive and the negative class [23]. These facts are depicted in Fig. 1.a and 1.b.

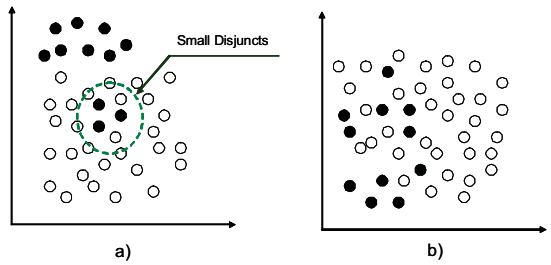

Figure 1: Example of the imbalance between classes: a) small disjuncts b) overlapping between classes

In this contribution we will focus on the data-sets with a higher degree of imbalance, using the imbalance ratio (IR) [24] as a threshold to categorize the different imbalanced scenarios. This measure is defined as the ratio of the number of instances of the majority class and the minority class.

In our previous work on this topic [25], 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 [11]. According to this, we will employ in this contribution the SMOTE algorithm in order to deal with the problem of 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.

Regarding the empirical measure, instead of using accuracy, a more correct metric is considered. This is due to the fact that accuracy can lead to erroneous conclusions, since it does not take into account the proportion of examples for each class. Because of this, in this work we use the AUC metric [9], which can be defined as

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

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 GP-COACH Algorithm

In this work we will make use of GP-COACH, a new FRBCS proposal [3]. The main features of this approach are listed below:

- It uses a context-free grammar that allows the learning of DNF fuzzy rules and the absence of some input features.
- It follows the *cooperative-competitive* approach, that is, it encodes a single rule per individual and the rule base (RB) is formed by the whole population. That makes necessary the use two different fitness functions in GP-COACH:
  - On the one hand, a local fitness function that evaluates the goodness of each one of the different rules in the population of individuals. From now on, we will refer it simply as *fitness function*.
  - On the other hand, a global fitness function that evaluates the goodness of a whole population of individuals (a rule set). From now on, we will refer it as *global fitness score*.

    This last fitness function has been introduced in GP-COACH in order to obtain the best rule set generated during the evolutionary process.
- It includes a mechanism to promote the diversity into the population, in order to avoid that all individuals converge to the same area of search space. Specifically, it uses the *Token Competition* diversity mechanism which makes rules compete among themselves during the evolutionary process, deleting irrelevant rules, and thus giving out a smaller number of rules that present a high-generalization capability.
- Finally, GP-COACH uses a two level hierarchical inference process because it learns rule sets containing two different types of rules: *primary rules*, which are strong and general rules generated by the genetic operators, and *secondary rules*, which are weaker and more specific rules, generated after the token competition procedure to increase the diversity in the population.

In the following subsections we will explain each one of these components and we will describe the way of working of this algorithm.

3.1 Context-free grammar for learning DNF fuzzy rules

GP-COACH learns DNF fuzzy rules:

$$R_k: \quad If\ X_1\ is\ \hat{A}_{k1}\ and\ \ldots\ and\ X_{n_v}\ is\ \hat{A}_{kn_v} \quad then\ Class\ is\ C_k\ with\ RW_k \tag{2}$$

where each input variable X_i takes as a value a set of linguistic terms or labels $\hat{A}_{ki} = \left\{ L_i^1\ or\ \ldots\ or\ L_i^{l_i} \right\}$ joined by a disjunctive operator, $(Class)$ is one of the class labels and RW_k is the rule weight [26]. We use triangular membership functions as antecedent fuzzy sets.

In GP-COACH, these DNF fuzzy rules are generated according to the production rules of a context-free grammar. In Table 1, an example of the grammar for a classification problem with two features (X_1, X_2), three linguistic labels per feature (*Low*, *Medium*, *High*) and three classes (C_1, C_2, C_3) is shown.

3.2 Evaluating an individual: Fitness function

Each one of the individuals in the population is evaluated according to a fitness function based on the estimation of:

ISBN: 978-989-95079-6-8

Table 1: Grammar example

$Start \longrightarrow [If],\ antec,\ [then],\ conseq,\ [.].$
$antec \longrightarrow descriptor1,\ [and],\ descriptor2.$
$descriptor1 \longrightarrow [any].$
$descriptor1 \longrightarrow [X_1\ is]\ label.$
$descriptor2 \longrightarrow [any].$
$descriptor2 \longrightarrow [X_2\ is]\ label.$
$label \longrightarrow \{member(?a,\ [L,\ M,\ H,\ L\ or\ M,\ L\ or\ H,$ $M\ or\ H,\ L\ or\ M\ or\ H])\},\ [?a].$
$conseq \longrightarrow [Class\ is]\ descriptorClass.$
$descriptorClass \longrightarrow \{member(?a,\ [C_1,\ C_2,\ C_3])\},\ [?a].$

- *Confidence*, which measures the accuracy of an individual, that is, the confidence of the consequent to be true if the antecedent is verified

$$Conf(R_k) = \frac{\mu_{tp}}{(\mu_{tp} + \mu_{fp})} \tag{3}$$

- *Support*, which measures the coverage of the knowledge represented in the individual

$$Supp(R_k) = \frac{\mu_{tp}}{N_{C_k}} \tag{4}$$

where μ_{tp} and μ_{fp} are the sums of the matching degrees for true and false positives, and N_{C_k} is the number of examples belonging to the class indicated in the consequent of the individual (R_k).

Both measures are combined to make up the fitness function in the following way

$$raw\_fitness = (\alpha * Conf) + ((1 - \alpha) * Supp) \tag{5}$$

where α parameter allow us to give more importance to any of these both measures.

3.3 Evaluating a population: Global fitness score

Global fitness score measure is defined as follows:

$$Global\_fitness = \begin{array}{l} (w_1 * AUC_{Tra}) + (w_2 * \overline{\#V}) + \\ (w_3 * \overline{\#C}) + (w_4 * \overline{\#R}) \end{array} \tag{6}$$

The global fitness score measure is formed by other four measures: a) AUC_{Tra}, the normalized value of the AUC metric for the training examples, b) *#V*, the normalized value of the number of variables per individual (rule) in the population, c) *#C*, the normalized value of the number of labels (or conditions) per individual, and d) *#R*, the normalized value of the number of rules in the population. We must remark that in the previous formula we employ the complement of these values.

Furthermore, we have included some weights (w_i) to give more importance to any of these four measures.

3.4 Token competition: Maintaining the diversity of the population

The idea of this mechanism is that each example in the training set can provide a resource called "token", for which all chromosomes in the population will compete to capture. If an individual (i.e. a rule) can match the example, it sets a flag to indicate that the token is seized preventing other weaker individuals to get this token.

The priority of receiving tokens is determined by the strength of the individuals. The individuals with higher fitness scores will exploit their niches by seizing as many tokens as they can. The other ones entering the same niches will have their strength decreased because they cannot compete with the stronger ones. This is done introducing a penalization in the fitness score of each individual. This penalization is based on the number of tokens that each individual has seized:

$$Penalized\_fitness = raw\_fitness * \frac{count}{ideal} \tag{7}$$

where *raw_fitness* is the fitness score obtained from the evaluation function, *count* is the number of tokens that the individual actually seized and *ideal* is the total number of tokens that it can take, which is equal to the number of examples that the individual matches.

As a result of the token competition, there exist individuals that cannot grab any token. These individuals are considered as irrelevant, and they can be eliminated from the population due to all of their examples are covered by other stronger individuals.

3.5 Secondary rules: Improving population diversity

Once the token competition mechanism has finished, it is possible that some training examples remain uncovered. The generation of new specific rules covering these examples improves the diversity in the population, and helps the evolutionary process to easily find stronger and more general rules covering these examples.

Therefore, GP-COACH learns rule sets having two different types of fuzzy rules: A core of strong and general rules (*primary rules*) that covers most of the examples, and a small set of weaker and more specific rules (*secondary rules*) that are only taken into account if there not exist any primary rule matching with some of the examples. This two level hierarchical inference process allows GP-COACH to obtain rule sets having a better interpretability-accuracy trade-off.

3.6 Genetic operators

GP-COACH makes use of four different genetic operators to generate new individuals during the evolutionary process:

1. *Crossover*: A part in the first parent is randomly selected and exchanged by another part, randomly selected, in the second one.
2. *Mutation*: It operates on label sets level. A variable in the rule is randomly chosen and then one of the next three different actions is carried out:
   (a) A new label is added to the label set.
   (b) A label is removed from the label set.
   (c) A label in the label set is exchanged by another one not included in it.
3. *Insertion*: It looks for all the variables in the rule and adds another different one that has not been included in it with a linguistic label set randomly chosen, although it must have at least one label and it must be different from the "any" set (see Table 1).
4. *Dropping Condition*: It randomly selects one variable in the rule and then turns it into "any". The label set associated with this variable is also removed.

We must remark that we generate only one child by using the genetic operators described above.

ISBN: 978-989-95079-6-8

3.7 Description of the Algorithm

GP-COACH algorithm begins creating a random initial population according to the rules in the context-free grammar. Each individual in this population is then evaluated. After that, the initial population is kept as the best evolved population and its global fitness score is calculated. Then, the initial population is copied to the current population and the evolutionary process begins:

1. An offspring population, with the same size than the current one, is created. Parents are selected by using the binary tournament selection mechanism and children are created by using one of the four genetic operators. The genetic operator selection is done in a probabilistic way. Specifically, the probabilities of the four genetic operator are added in a single measure $P = P_c + P_m + P_i + P_{dp}$ (where P_c, P_m, P_i and P_{dp} represent the crossover, mutation, insertion and dropping condition probabilities, respectively) and then a random value $u \in [0, P]$ is obtained to choose the genetic operator to be applied.

2. Once the offspring population is created, it is joined to the current population, creating a new population whose size is double the current population size. Individuals in this new population are sorted according to their fitness and Token Competition mechanism is applied. Secondaries rules are created if some examples remain uncovered.

3. Global fitness score measure is then calculated for this new population. We check whether this new fitness is better than the one stored for the best population, updating the best population and fitness if necessary. In any case, the new population is copied as the current population in order to be able to apply the evolutionary process again.

The evolutionary process ends when the stop condition is verified (i.e. number of evaluations). Then, the population kept as the best one is returned as a solution to the problem and GP-COACH finishes.

4 Experimental Study

In this study, our aim is to analyse the behaviour of the GP-COACH approach in the context of data-sets with high imbalance. We will compare the performance of this method against one state-of-the-art FRBCS algorithm, the Chi et al.'s approach, and the C4.5 decision tree, employing a large collection of imbalanced data-sets.

Specifically, we have considered twenty-two data-sets from UCI repository [10] 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 ascending order according to the IR. Data-sets with more than two classes have been modified by taking one against the others or by contrasting one class with another.

In order to reduce the effect of imbalance, we will employ the SMOTE preprocessing method [11] for all our experiments, considering only the 1-nearest neighbour to generate the synthetic samples, and balancing both classes to the 50% distribution.

In the remaining of this section, we will first present the experimental framework and all the parameters employed in this study and then we will show the results and all the statistical study for the GP-COACH approach.

4.1 Experimental Framework

To develop the different experiments we consider a *5-fold 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 GP-COACH is a probabilistic method, we perform three executions per partition with different random seeds. For each data-set we consider the average results of the five partitions per three executions. Furthermore, Wilcoxon's Signed-Ranks Test [27] is used for statistical comparison of our empirical results. In all cases the level of confidence (α) will be set at 0.05.

The configuration for the FRBCSs approaches, GP-COACH and Chi et al.'s, is presented in Table 3. This parameter selection has been carried out according to the results achieved by GP-COACH and Chi et al.'s in our former studies in [3] and [25] respectively.

Table 3: Configuration for the FRBCS approaches

Parameter	GP-COACH	Chi et al.'s
Conjunction operator	Minimum T-norm	Product T-norm
Rule Weight	Certainty Factor	Penalized Certainty Factor
Fuzzy Reasoning Method	Additive Combination	Winning Rule
Number of Labels	5 Labels	5 Labels

The specific parameters setting for the GP-COACH algorithm, is listed below:

- Number of evaluations: 20000 evaluations.
- Initial Population Size: 200 individuals.
- α: 0.7.
- Crossover Probability P_c: 0.5.
- Mutation Probability P_m: 0.2.
- Dropping Probability P_{dp}: 0.15.
- Insertion Probability P_i: 0.15.
- Tournament Size: 2.
- w_1: 0.8, w_2 = w_3: 0.05, w_4: 0.1.

4.2 Analysis of the GP-COACH Behaviour in Data-sets with High Imbalance

The main objective in this study is to analyse the behaviour of the GP-COACH approach in the context of data-sets with high imbalance. According to this, Table 4 shows the results in performance (using the AUC metric) for GP-COACH and the algorithms employed for comparison, that is, Chi et al.'s learning method and C4.5.

We observe that the performance obtained by GP-COACH is higher than the one for Chi et al.'s and C4.5. This situation is represented statistically by means of a Wilcoxon test (Table 5) which shows a higher ranking in both cases for the GP-COACH algorithm.

Furthermore, when we compare the results for the fuzzy methods in each data-set, we observe that, as the IR increases, GP-COACH achieves a better performance than Chi et al.'s method. In this manner, we may state that GP-COACH is a very robust method according to the IR.

ISBN: 978-989-95079-6-8

Table 2: Summary Description for Imbalanced Data-Sets.

Data-set	#Ex.	#Atts.	Class (min.; maj.)	%Class(min., maj.)	IR
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	(nuc; vac)	(6.72, 93.28)	13.87
Shuttle0vs4	1829	9	(Rad Flow; Bypass)	(6.72, 93.28)	13.87
Glass4	214	9	(containers; remainder)	(6.07, 93.93)	15.47
Page-blocks13vs2	472	10	(graphic; horiz.line,picture)	(5.93, 94.07)	15.85
Abalone9vs18	731	8	(18; 9)	(5.65, 94.25)	16.68
Glass016vs5	184	9	(tableware; build-win-float-proc, build-win-non_float-proc,headlamps)	(4.89, 95.11)	19.44
Shuttle2vs4	129	9	(Fpv Open; Bypass)	(4.65, 95.35)	20.5
Yeast1458vs7	693	8	(vac; nuc,me2,me3,pox)	(4.33, 95.67)	22.10
Glass5	214	9	(tableware; remainder)	(4.20, 95.80)	22.81
Yeast2vs8	482	8	(pox; cyt)	(4.15, 95.85)	23.10
Yeast4	1484	8	(me2; remainder)	(3.43, 96.57)	28.41
Yeast1289vs7	947	8	(vac; nuc,cyt,pox,erl)	(3.17, 96.83)	30.56
Yeast5	1484	8	(me1; remainder)	(2.96, 97.04)	32.78
Ecoli0137vs26	281	7	(pp,imL; cp,im,imU,imS)	(2.49, 97.51)	39.15
Yeast6	1484	8	(exc; remainder)	(2.49, 97.51)	39.15
Abalone19	4174	8	(19; remainder)	(0.77, 99.23)	128.87

Table 4: Detailed results in performance (AUC metric) for GP-COACH, Chi et al.'s learning method and C4.5

Dataset	GP-COACH		Chi et al.'s		C4.5	
	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
Yeast2vs4	83.46 ± 2.01	78.26 ± 5.92	90.51 ± 1.43	86.85 ± 6.68	98.14 ± 0.88	85.88 ± 8.78
Yeast05679vs4	82.76 ± 1.25	79.92 ± 5.54	87.97 ± 0.65	76.42 ± 6.17	95.26 ± 0.94	76.02 ± 9.36
Vowel0	96.74 ± 0.76	92.20 ± 5.80	99.64 ± 0.19	97.89 ± 1.83	99.67 ± 0.48	94.94 ± 4.95
Glass016vs2	74.57 ± 2.98	59.35 ± 17.29	76.16 ± 2.11	60.02 ± 8.41	97.16 ± 1.86	60.62 ± 12.66
Glass2	78.45 ± 2.82	66.11 ± 13.76	75.50 ± 1.80	52.06 ± 11.20	95.71 ± 1.51	54.24 ± 14.01
Ecoli4	97.74 ± 0.66	91.91 ± 6.04	98.14 ± 0.65	92.30 ± 8.13	97.69 ± 1.96	83.10 ± 9.90
Yeast1vs7	76.44 ± 2.17	64.03 ± 8.50	84.08 ± 2.14	65.24 ± 10.47	93.51 ± 2.20	70.03 ± 1.46
Shuttle0vs4	99.88 ± 0.08	99.99 ± 0.04	100.0 ± 0.00	98.72 ± 1.17	99.99 ± 0.02	99.97 ± 0.07
Glass4	95.94 ± 1.87	86.94 ± 14.19	98.88 ± 0.56	82.85 ± 10.20	98.44 ± 2.29	85.08 ± 9.35
Page-Blocks13vs4	98.03 ± 0.82	96.39 ± 4.31	98.71 ± 0.23	93.41 ± 8.53	99.75 ± 0.21	99.55 ± 0.47
Abalone9vs18	77.91 ± 2.31	74.27 ± 7.15	71.22 ± 3.09	67.44 ± 9.88	95.31 ± 4.44	62.15 ± 4.96
Glass016vs5	95.83 ± 1.45	94.29 ± 8.21	98.43 ± 0.41	84.86 ± 21.91	99.21 ± 0.47	81.29 ± 24.44
Shuttle2vs4	97.36 ± 3.34	97.43 ± 3.78	100.0 ± 0.00	88.38 ± 21.60	99.90 ± 0.23	99.17 ± 1.86
Yeast1458vs7	66.36 ± 1.61	58.21 ± 8.47	81.83 ± 1.70	59.32 ± 7.68	91.58 ± 2.78	53.67 ± 2.09
Glass5	98.11 ± 1.01	78.05 ± 24.24	98.78 ± 0.48	74.63 ± 20.52	99.76 ± 0.40	88.29 ± 13.31
Yeast2vs8	80.97 ± 3.28	78.77 ± 9.37	83.46 ± 1.68	80.66 ± 6.94	91.25 ± 1.84	80.66 ± 11.22
Yeast4	84.88 ± 1.42	81.95 ± 4.08	87.96 ± 1.54	83.25 ± 2.39	91.01 ± 2.64	70.04 ± 5.65
Yeast1289vs7	72.17 ± 2.75	66.26 ± 10.42	80.03 ± 2.33	70.27 ± 3.75	94.65 ± 1.13	68.32 ± 6.16
Yeast5	95.78 ± 0.68	93.53 ± 2.74	95.43 ± 0.54	93.72 ± 2.72	97.77 ± 1.45	92.33 ± 4.72
Ecoli0137vs26	90.13 ± 2.42	81.00 ± 18.30	96.85 ± 1.59	68.80 ± 22.87	96.78 ± 3.28	81.36 ± 21.68
Yeast6	90.43 ± 1.68	86.67 ± 7.92	89.60 ± 2.00	88.20 ± 8.55	92.42 ± 3.54	82.80 ± 12.77
Abalone19	74.00 ± 3.70	68.45 ± 8.38	77.19 ± 2.49	67.48 ± 10.77	85.44 ± 2.49	52.02 ± 4.41
Mean	86.72 ± 1.87	**80.64 ± 8.84**	89.56 ± 1.25	78.76 ± 9.65	95.93 ± 1.68	78.25 ± 8.38

Table 5: Wilcoxon test to compare GP-COACH with Chi et al.'s approach and C4.5 according to their performance. R^+ corresponds to GP-COACH and R^- to Chi or C4.5

Comparison	R^+	R^-	Hypothesis ($\alpha = 0.05$)	p-value
GP-COACH vs. Chi	161.0	92.0	Not Rejected	0.263
GP-COACH vs. C4.5	162.0	91.0	Not Rejected	0.249

Regarding interpretability of the obtained models, we must stress that GP-COACH is designed to obtain few DNF rules to describe the concept accurately. Table 6 shows the average number of rules for the three algorithms considered in this study, together with the associated standard deviation. The results from this table clearly shows that GP-COACH is the most interpretable model.

In Table 7 we show an example of an RB generated with the GP-COACH algorithm for the "shuttle2vs4" data-set. We can observe that the problem is described using few rules and only three of nine variables, enhancing the readability for the end-user.

5 Conclusions

In this contribution, we have studied the behaviour of GP-COACH in the context of data-sets with high imbalance.

Our results have shown the good performance achieved by this approach in contrast with the Chi et al.'s method, a well-known fuzzy rule learning algorithm, and C4.5, an algorithm of reference in the area of imbalanced data-sets.

Furthermore, we have compared the interpretability of the obtained models by means of the size of the rule set, concluding that GP-COACH employs a more compact RB in comparison with Chi et al.'s and C4.5 algorithms.

We must remark that GP-COACH is a good methodology for imbalanced data-sets, since it obtains very good classification results according to the AUC measure, by using a very interpretable model with few linguistic fuzzy rules.

Acknowledgment

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

01 and TIN2008-06681-C06-02.

Table 6: Detailed results table in interpretability (number of rules) for GP-COACH, Chi et al.'s learning method and C4.5

Data-set	GP-COACH	Chi et al.'s	C4.5
Yeast2vs4	6.60 ± 0.74	164.80 ± 3.83	20.40 ± 4.56
Yeast05679vs4	9.87 ± 2.17	191.20 ± 9.12	30.00 ± 6.63
Vowel0	5.60 ± 1.12	694.60 ± 8.32	11.80 ± 2.17
Glass016vs2	6.60 ± 2.23	65.20 ± 5.67	15.20 ± 1.48
Glass2	5.53 ± 1.96	73.20 ± 2.28	16.80 ± 2.49
Ecoli4	5.67 ± 0.90	116.80 ± 6.42	9.20 ± 2.17
Yeast1vs7	8.80 ± 2.01	156.80 ± 6.46	33.20 ± 6.10
Shuttle0vs4	2.60 ± 0.63	79.20 ± 7.56	2.80 ± 1.10
Glass4	6.33 ± 1.59	98.40 ± 10.06	7.40 ± 2.88
Page-Blocks13vs4	5.07 ± 0.96	164.00 ± 7.78	7.00 ± 1.22
Abalone9vs18	9.20 ± 2.24	93.60 ± 5.03	47.60 ± 1.14
Glass016vs5	5.13 ± 0.74	99.60 ± 8.62	10.00 ± 2.83
Shuttle2vs4	3.60 ± 0.99	33.00 ± 4.95	4.00 ± 0.00
Yeast1289vs7	8.93 ± 1.91	160.80 ± 2.68	58.40 ± 6.35
Glass5	4.93 ± 0.70	90.80 ± 6.87	7.00 ± 1.00
Yeast2vs8	5.73 ± 1.53	110.00 ± 2.65	14.60 ± 2.07
Yeast4	8.07 ± 1.22	197.20 ± 7.01	40.40 ± 3.78
Yeast1458vs7	7.27 ± 1.22	181.00 ± 8.69	47.80 ± 6.38
Yeast5	3.47 ± 0.52	206.60 ± 5.32	11.60 ± 2.07
Ecoli0137vs26	5.13 ± 0.74	168.60 ± 5.41	8.00 ± 1.41
Yeast6	5.73 ± 1.49	198.80 ± 6.14	21.60 ± 6.47
Abalone19	6.80 ± 1.15	180.20 ± 7.40	69.20 ± 3.70
Mean	**6.21 ± 1.31**	160.20 ± 6.29	22.45 ± 3.09

Table 7: Example of a DNF RB extracted using GP-COACH for the shuttle2vs4 data-set

Rule1: IF X_7 is $(L_4|L_5)$ THEN Class = negative with RW = 1
Rule2: IF X_1 is $(L_3|L_4|L_5)$ AND X_7 is $(L_1|L_2)$
THEN Class = positive with RW = 0.934089
Rule3: IF X_7 is $(L_2|L_4)$ AND X_8 is L_4
THEN Class = positive with RW = 0.495765
Rule4: IF X_7 is $(L_3|L_5)$ THEN Class = positive
with RW = 0.562457

References

[1] Q. Yang and X. Wu. 10 challenging problems in data mining research. *International Journal of Information Technology and Decision Making*, 5(4):597–604, 2006.

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

[3] F.J. Berlanga, M.J. del Jesus, and F. Herrera. A novel genetic cooperative-competitive fuzzy rule based learning method using genetic programming for high dimensional problems. In *3rd International Workshop on Genetic and Evolving Fuzzy Systems (GEFS08)*, pages 101–106, 2008.

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

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

[6] A. Estabrooks, T. Jo, and N. Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1):18–36, 2004.

[7] C.T. Su and Y.H. Hsiao. An evaluation of the robustness of MTS for imbalanced data. *IEEE Transactions on Knowledge Data Engineering*, 19(10):1321–1332, 2007.

[8] J. Casillas, F. Herrera, R. Pérez, M.J. del Jesus, and Pedro Villar. Special issue on genetic fuzzy systems and the interpretability-accuracy trade-off - editorial. *International Journal of Approximate Reasoning*, 44(1):1–3, 2007.

[9] J. Huang and C. X. Ling. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005.

[10] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. University of California, Irvine, School of Information and Computer Sciences. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligent Research*, 16:321–357, 2002.

[12] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[13] 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*, 9:2607–2624, 2008.

[14] 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*, 13:10 959–977, 2009.

[15] N. V. Chawla, N. Japkowicz, and A. Kolcz. Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explorations*, 6(1):1–6, 2004.

[16] R. Barandela, J.S. Sánchez, V. García, and E. Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851, 2003.

[17] 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*, 178(16):3214–3227, 2008.

[18] T. Fawcett and F. J. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.

[19] 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*, 7(4):720–747, 2006.

[20] M.A. Mazurowski, P.A. Habas, J.M. Zurada, J.Y. Lo, J.A. Baker, and G.D. Tourassi. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*, 21(2-3):427–436, 2008.

[21] G. M. Weiss. Mining with rarity: a unifying framework. *SIGKDD Explorations*, 6(1):7–19, 2004.

[22] G.M. Weiss and F. Provost. Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, 19:315–354, 2003.

[23] V. García, R.A. Mollineda, and J. S. Sánchez. On the k-NN performance in a challenging scenario of imbalance and overlapping. *Pattern Analysis Applications*, 11(3–4):269–280, 2008.

[24] A. Orriols-Puig and E. Bernadó-Mansilla. Evolutionary rule-based systems for imbalanced datasets. *Soft Computing*, 13(3):213–225, 2009.

[25] 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*, 159(18):2378–2398, 2008.

[26] H. Ishibuchi and T. Yamamoto, Rule Weight Specification in Fuzzy Rule-Based Classification Systems. *IEEE Transactions on Fuzzy Systems*, 13(4):428-435, 2005.

[27] D. Sheskin. *Handbook of parametric and nonparametric statistical procedures*. Chapman & Hall/CRC, second edition, 2006.

ISBN: 978-989-95079-6-8