2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Designing a compact Genetic Fuzzy Rule-Based System for One-Class Classification

Pedro Villar, Bartosz Krawczyk, Ana M. Sánchez, Rosana Montes and Francisco Herrera

***Abstract*— This paper proposes a method for designing Fuzzy Rule-Based Classification Systems to deal with One-Class Classification, where during the training phase we have access only to objects originating from a single class. However, the trained model must be prepared to deal with new, unseen adversarial objects, known as outliers. We use a Genetic Algorithm for learning the granularity, domains and fuzzy partitions of the model and we propose an ad-hoc rule generation method specific for One-Class Classification. Several datasets from UCI repository, previously transformed to one-class problems, are used in the experiments and we compare with two of the classical methods used in the One-Class community, one-class Support Vector Machines and Support Vector Data Description. Our proposal of fuzzy model obtains similar results than the other methods but presents a high interpretability due its reduced number of rules.**

I. Introduction

In Machine Learning, traditional methods try to classify a new element considering a discrete set of categories or classes. In the One-Class Classification (OCC) paradigm[1][2], one of these categories is sufficiently described in the examples of the training data and usually it is named positive class, objective class or target class. However, for the rest of classes, those represent the negative concept, or simply the no belonging to the objective class, there are no examples or there are so few of them (not sufficient for characterize that concept).

In this contribution, we use Fuzzy Rule-Based Classification Systems (FRBCSs), that present two main components: the Inference System and the Knowledge Base (KB). The KB is composed of the Rule Base (RB) constituted by the set 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 a FRBCS directly depends on the problem being solved. If some expert information about the problem under solving is not available, it is possible to generate the KB from examples by an automatic learning process.

In previous works[3][4], we have demonstrated the high influence of the DB design in the behavior of a Fuzzy Rule-Based System for regression or classification and we proposed a methodology of learning that evolves DB definitions by means of a Genetic Algorithm (GA) that includes a RB generation method in the evaluation process of each individual. In the specialized literature, there are several proposal for learning the KB of an FRBCS, or for learning/adjusting some of its components. However, the majority of these methods are designed for multi-class problems, specially for learning the RB. Therefore, in this paper we propose a new RB generation method specific for OCC that try to get a reduced set of rules for describing the objective class and also we propose a simple fuzzy reasoning method for OCC that allows to determine if a new example is classified or not as belonging to the objective class.

In this paper we use a Genetic Algorithm for learning the main components of the DB: the granularity level (number of labels) of each variable, the domain of each variable (we allow a slight extension of the domain in the two extremes) and the form of each fuzzy membership function in non-uniform fuzzy partitions, with different areas in the variable working range where the partition has a higher or a lower relative sensibility. The GA learns the complete BC using the RB generation method mentioned before.

As regards to the experimental study, we have followed the same idea than other works[5][6] [7] and we have used binary-class datasets that are converted to one-class datasets considering one of the classes as objective class and the other as anomalous class (or simply ”not objective” class). Therefore, the training data set only contains examples of the objective class while the test data set contains examples of both classes. We compare our proposal with two well known methods in the OCC community, the One-Class Support Vector Machines (OCSVM)[8] and Support Vector Data Description (SVDD)[9]. Due to the reduced number of rules of the FRBCS obtained, the interpretability of the proposed model is very high in opposite to the models obtained by OCSVM and SVDD.

This paper is organized as follows. First, Section II introduces the preliminary concepts of OCC, emphasizing its differences with traditional multi-class classification. Next, in Section III we describe the RB generation method and the fuzzy reasoning method used in the learning process while in Section IV we will expose the main characteristics of our learning process: a GA for designing the KB of a FRBCS for OCC. The next section describes the experimental study. Finally, in Section VI, some conclusions will be pointed out.

P. Villar, A. M. Sánchez and R. Montes are with the Department of Software Engineering, University of Granada, Spain; B. Krawczyk is with the Department of Systems and Computer Networks, Wrocław University of Technology, Poland; F. Herrera is with the Department of Computer Science and A. I., University of Granada, Spain; (email: {pvillarc, amlopez, rosana}@ugr.es , bartosz.krawczyk@pwr.wroc.pl, herrera@decsai.ugr.es).

This work was partially supported by the Spanish Ministry of Science and Technology under project TIN2011-28488 and the andalusian regional projects P10-TIC-06858 and P11-TIC-9704. Bartosz Krawczyk was supported by The Polish National Science Centre under the grant PRELUDIUM number DEC-2013/09/N/ST6/03504

II. One-class Classification

The term one-class classification was introduced in[10], although several authors have used other terms for designating similar paradigms like outliers detection, anomaly detection, novelty detection or concept learning. This variability of names is more related with the concrete applications in which the paradigm is performed than the differences in the models used for representing the problem or the way of designing these models.

OCC has found many real-life applications. Examples are intrusion detection, text classification[11], medical analysis[12], bioinformatics[13] or spam detection[14].

What is distinctive to OCC is the classifier learning phase. In conventional classification problems, a learner is trained in such a way as to minimize the recognition error between objects belonging to two or more classes. This is not valid for OCC. The most important paradigm of one-class learning is that during the classifier training stage, it has only an access to objects coming from a single class (target concept class). Outliers appear during the exploitation phase, but are unknown at the training step. Therefore, minimizing the error on just a single class would easily lead to an overfitting of the model.

Problems, that are encountered in multi-class classification (such as: reducing the classification mistakes, improving the generalization capability of the model, dealing with the curse of dimensionality, etc.) are in most cases still valid for OCC [16]. However, new problems are also encountered, such as consistency of rejection rate or empty sphere in competence areas [17].

OCC aims at building a model, which has a strong discriminant properties against outliers, and at the same time can properly accept new objects fulfilling the assumptions of the target class. In this way OCC learner should have good trade-off between discriminative powers and generalization abilities. For example, the definition of the classification boundary of a class is more difficult without examples of other classes, that is, how far from the data should be the boundary. If it is very near of the data, some examples belonging to the class can be misclassified. On the other hand, if the boundary is a bit far of the data, some examples out to the class can be misclassified as belonging to the class.

There are four main approaches for OCC:

- Density-based methods, which aim at estimating a complete density of the target class. They are simple, yet efficient - however require a large size of the training set to work [18].
- Reconstruction-based methods, which aim at detecting structure or topology in the target class [19].
- Boundary-based methods, which aim at forming an enclosing boundary around the target class in hope that it will sufficiently describe the target concept. Among them Support Vector proposals are the most popular and we have chosen two classical methods for comparing with our proposal: One-Class Support Vector Machines (OCSVM)[8] and Support Vector Data Description (SVDD)[9].
- Ensemble methods, that work on the basis of utilizing more than one predictor and combining their local diverse competencies [20] [21].

It should be noted, that so far there are no dedicated benchmarks for OCC. Therefore, there is a need for using standard binary and multi-class benchmarks, and changing their nature to suit OCC. So far, there are several (quite similar) proposals on how to conduct this procedure [5][6] [7].

Most common approaches can be described as follows:

1) If the number of classes are greater than two, the dataset is transformed in a binary-class problem. The two new classes are defined as the joint of one or more previous classes.
2) Build the training and test data sets. The training data set only contains examples of one of the two classes (objective or target class). The test data set contains the remaining examples of the objective class and all the examples belonging to the other class.

Many proposals for learning classifiers use some kind of accuracy measure like the accuracy over the example set. However, these measures can lead to erroneous conclusions working with imbalanced datasets since it doesn't take into account the proportion of examples for each class. In our experimental study, we use binary classification datasets with low degree of imbalance. So, that situation can appear in the analysis of the test data set performance. Therefore, in this work, for evaluating the performance in the test data set, we use the Area Under the Curve (AUC) metric [22], 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 (objective class in our case) and FP_{rate} is the percentage of negative cases (examples of the other class) misclassified as belonging to the positive class. Other OCC works[5][6] also use the AUC metric for evaluating the model performance.

III. Determination of the Rule Base and classification of new examples

In this section we will describe our Rule Base generation method, specific for OCC and the fuzzy reasoning method used to classify a new example.

A. Rule Base generation algorithm

In this work, considering a problem with N variables, the rules of the FRBCS have the following structure:

$$\begin{array}{l} \text{Rule } R_j : \text{ If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn} \\ \text{then Class} = C \end{array} \tag{2}$$

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, and C is the objective class label. We use triangular membership functions as antecedent fuzzy sets.

The RB generation method tries to get a compact set of rules, that fine tune the examples of the objective class presented in the training data set, by selecting a reduced number of rules that cover so many examples each one of them. The algorithm only determines the antecedents of the rules as the consequent is the same for all the rules (the objective class).

The operation of the algorithm is the following. For each example of the training data set (all of them belonging to the objective class), the next steps are performed:

1) Select the best possible antecedent for that example, looking for the label with high membership degree for each component.
2) Check if that antecedent has been explored before. If it is a new antecedent, the decision of including that rule in the RB depends on the value of two measures:
   - **covering average**: the average of the matching degrees of the rule for all the examples with matching degree for that rule greater than zero. To calculate the matching degree of an example with the antecedent of the rule, product t-norm is used.
   - **support**: the number of examples covered by the rule (matching degree greater than zero) divided by the total number of training examples.

   The purpose of using these measures is to generate a compact set of rules covering the majority of examples of the training data set. Therefore, the antecedent generated is accepted as a rule if it fulfills the next two requirements:
   - **covering average** $> threshold\_covering$
   - **support** $> threshold\_support$

   where $threshold\_covering$ and $threshold\_support$ are parameters of the method with values in the interval $[0, 1]$.

B. Fuzzy Reasoning Method for classifying new examples

Once the RB has been generated, the reasoning mechanism to determine if a new example is classified or not as the objective class is composed of three steps:

1) Select the rules with matching degree greater than zero for that example.
2) Calculate the average of the matching degree obtained for the rules selected in the previous step.
3) If that average value is greater than $threshold\_rule$, the example is classified as belonging to the objective class. In the other case, the example is classified as "anomalous" or simply considered as not belonging to the objective class ($threshold\_rule$ is a parameter of the method with values in the interval $[0, 1]$).

IV. Genetic Algorithm for FRBCS Learning

In this section, we describe the learning approach to automatically generate the KB of an FRBCS, that is composed of two methods:

- A genetic learning process for the components of the DB:
  – The number of labels for each linguistic variable (granularity level).
  – The variable domain (working range), allowing a brief extension of the domain in the two extremes.
  – The form of each fuzzy membership function in non-uniform fuzzy partitions, using a non-linear scaling function that defines different areas in the variable domain where the partition has more joint labels (higher sensibility) or more separate labels (lower sensibility).
- The RB generation method presented in SectionIII-A.

The granularity level per variable has a great influence in the final model performance, as stated in [15] for regression problems. Our GA also evolves the variable working range, usually not considered in the learning algorithms, and it is based on one of the methods proposed in [3] for regression problems.

We denote our proposal as FRS-OC-GA (Fuzzy Rule-based System for One-Class classification designed by a Genetic Algorithm).

Many components of the DB will be adapted throughout a GA. Since it is desirable to reduce the dimensionality of the search space for that process, the non-linear scaling functions should be parameterized functions with a reduced number of parameters. We have used the scaling function proposed in [3], that has a single sensibility parameter called a ($a \in \mathbb{R}$). The function used is ($f : [-1, 1] \rightarrow [-1, 1]$)

$$f(x) = sign(x) \cdot |x|^a, \quad with \;\; a > 0$$

The final result is a value in $[-1, 1]$. There are three different possibilities in sensibility of the fuzzy partition depending on the value of the parameter a. If ($a = 1$), the function produces uniform sensibility, that is, the typical uniform fuzzy partition. If $a > 1$, the function produces higher sensibility for center values while higher sensibility for extreme values are generated if $a < 1$. In this paper, triangular membership functions are considered. So, the non-linear scaling function will only be applied on the three definition points of the membership function. Figure 1 shows a graphical representation of these three possibilities of fuzzy partition depending on the value of parameter a.

We should note that the previous scaling function should be used with symmetrical attributes since it causes symmetrical effects around the center point of the interval. It can not produce higher sensibility in only one of the working range extents. In the method presented in this paper, we add a new parameter (called S), with only two possible values ($\{0, 1\}$) to the non-linear scaling function as described also in [3]. The parameter S has no effect when $S = 0$ and the fuzzy partition generation depends only on the value of parameter a as commented previously (Figure 1). On the other hand, fuzzy partitions with asymmetric shape are generated when $S = 1$ if the parameter a is not equal to 1 (lower sensibility for the lowest or for the highest values, Figure 2).

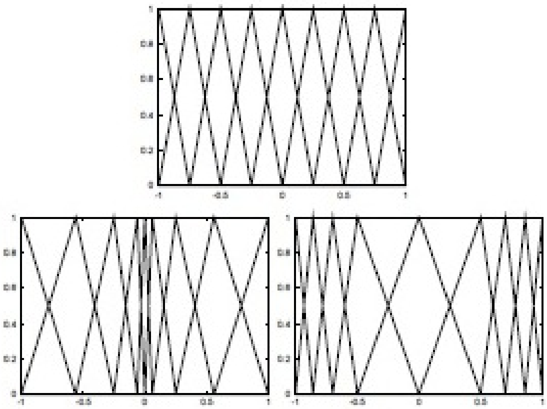

Fig. 1. Fuzzy partitions with $a = 1$ (top), $a > 1$ (down left), and $a < 1$ (down right)

The next subsections describe the main components of the genetic learning process.

A. Encoding the DB

The three components of the DB encoded in the chromosomes are the number of linguistic terms for variable, the membership functions that define their semantics and the working ranges.

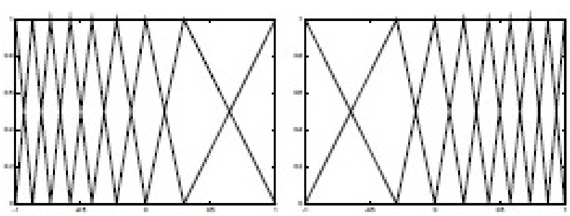

Fig. 2. Fuzzy partitions with $S = 1$ (left with $a > 1$ and right with $a < 1$)

As commented in the previous section, to generate the fuzzy partitions, we consider triangular-shaped functions symmetrical and uniformly distributed across the variable working range. Then, we apply the non-linear function with its sensibility parameters. Therefore, the domain, the granularity level and the sensibility parameters for each variable are enough to define the whole fuzzy partition.

Therefore, each chromosome will be composed of three parts:

- Granularity level (C_1): For a dataset with *N* attributes, the number of labels per variable is encoded into an array of length *N*. In this paper, the possible values considered are the set $\{2, \ldots, 7\}$.
- Sensibility parameters (C_2): An array of length $N \times 2$, where the sensibility parameters (*a,S*) are stored for each variable. In this contribution, the range considered for the parameter *a* is the interval $[0, 10)$.
- Working ranges (C_3): The variable domain ($[r^{inf}, r^{sup}]$) is encoded into an array of $N \times 2$ real values. If the initial working range of the variable *i* is $[v_i^{min}, v_i^{max}]$, and *d* is the interval dimension ($d = v_i^{max} - v_i^{min}$), the ranges considered for the variable domain limits are:
  lower limit: $[v_i^{min} - (1/4 * d), v_i^{min}]$
  upper limit: $[v_i^{max}, v_i^{max} + (1/4 * d)]$

Hence, the structure of each individual is summarized next (considering that $R_i = \{r_i^{inf}, r_i^{sup}\}$):

$$C_1 = (l_1, \ldots, l_N)$$

$$C_2 = (a_1, \ldots, a_N, S_1, \ldots, S_N)$$

$$C_3 = (R_1, \ldots, R_N)$$

$$C = C_1 C_2 C_3$$

B. Initial Gene Pool

The initial population can be divided in three parts, the first two contain $\#val \times 5$ chromosomes, with $\#val$ being the cardinality of the granularity values term set (in our case $\#val = 6$, corresponding to the six possibilities for the number of labels, $2 \ldots 7$). So, the number of chromosomes (M) has to be at least greater than $\#val \times 10$. The composition of the three parts is described next:

- The first $\#val \times 5$ chromosomes will have the same number of labels and the initial working range in all its variables. For each possible granularity level, five chromosomes with the main possibilities for the sensibility parameters will be created: one with $a = 1$, two with $a < 1$ (one with $S = 0$ and another with $S = 1$) and the other two with $a > 1$ (one with $S = 0$ and another with $S = 1$). The values of the parameter a are generated at random.
- The second $\#val \times 5$ chromosomes are equal to the first group, but randomly changing the variable working range. Each chromosome will have the same number of labels in all its variables. For each possible granularity level, five chromosomes are created as in the first part of the population with random values for the parameter a. For the third part of the chromosomes, two random values in the variable domain interval (lower and upper) are generated.
- In the remaining individuals ($M - (\#val \times 10)$ chromosomes), all the components are selected at random. In this paper, this part is comprised by 40 chromosomes, so, the population length is 100.

C. Evaluating the chromosome

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

1) Generate the fuzzy partitions using the information contained in the chromosome. First, each variable is linearly mapped from the working range codified in the chromosome ($[r^{inf}, r^{sup}]$) to interval $[-1, 1]$. In a second step, uniform fuzzy partitions for all the variables are created considering the number of labels per variable (l_i). Finally, the non-linear scaling function with its sensibility parameters (a_i, S_i) is applied to the three definition points of the membership functions

obtained in the previous step, obtaining the whole DB definition.

2) Generate the RB by running the algorithm described in SectionIII-A.

3) According with the fuzzy reasoning method presented in SectionIII-B, classify all the examples of the training data set. The fitness value will be the number of examples correctly classified, that is, classified as belonging to the objective class. The GA tries to maximize this fitness value in each generation.

D. Genetic operators

We use the Baker's stochastic universal sampling [23] as selection mechanism, in which the number of any structure offspring is limited by the lower and upper values of the expected number of descendants. We also employ an elitist scheme, maintaining the best individual in the next generation.

A set of genetic operators is applied to the individuals of one generation, to obtain the next generation. Due to the diversity of the information encoded in the chromosomes, the design of genetic operators should deal with it. As there is a strong relationship among the three chromosome parts, the operators selected must work cooperatively in C_1, C_2 and C_3 in order to make best use of the representation tackled. Taking into account these aspects, the following operators are considered:

1) Crossover: Two different crossover operators are considered depending on the two parents' scope:

- *Crossover when both parents have the same granularity level per variable:* If the two parents have the same values in C_1, it is probable that the genetic search has located a promising granularity level, that it should be adequately exploited. This task is developed by applying the max-min-arithmetical (MMA) crossover operator [24] in the chromosome parts based on real-coding scheme, that is, the parameters a_i (first part of C_2) and the working ranges (C_3). Obviously, the parent C_1 values are maintained in the offspring. The two possible values of parameter S_i are tested and the best two chromosomes are selected.
- *Crossover when the parents encode different granularity levels:* In this case, the usual scheme of the crossover operator is performed in order to discover new promising zones. So, a standard one-point crossover operator is applied over the three parts of the chromosome. This operator performs as follows: a crossover point p is randomly generated (with values between 2 and N) and the two parents are crossed at the p-th variable in all the chromosome parts, thereby producing two meaningful descendants.

2) Mutation: Three different operators are used, each one of them acting on different chromosome parts:

- *Mutation on C_1*: The mutation operator for the granularity levels is similar to the one proposed by Thrift in [25]. The number of labels of the variable is changed to the immediately upper or lower value (the decision is made at random). When the value to be changed is the lowest (2) or the highest one (7), the only possible change is developed.
- *Mutation on the second part of C_2 (parameters S_i)*: As that part of the chromosome is binary coded, a simple binary mutation is used, flipping the value of the gene.
- *Mutation on the first part of C_2 (parameters a_i) and C_3*: As this part is based on a real-coding scheme, Michalewicz's non-uniform mutation operator is employed [26].

V. Experimental study

We will study the performance of FRS-OC-GA employing 16 binary datasets with low degree of imbalance from KEEL dataset repository [27], which are publicly available on the corresponding web-page[1], including general information about them.

A possible value for measure the degree of imbalance of a dataset is the imbalance ratio (IR) [28], which is defined as the ratio of the number of instances of the majority class and the minority class. We consider that a dataset presents a low degree of imbalance when its IR is lower than 9.

Table I presents the datasets, where we show 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. With the aim of obtaining binary imbalanced problems, the two classes are defined as the joint of one or more classes, which are specified in column *Class* of Table I separated by a semi-colon.

In the KEEL dataset repository, there are available 5-folder cross-validation partitions for all these datasets. We use these partitions with a previous conversion to one-class problems as explained in the introduction. We consider the majority class as objective class an the minority class as "anomalous" class. Therefore, all the examples of the "anomalous" class presented in the training data set of each fold are moved to the correspondent test data set. So, the percentage of the number of examples in each training-test distribution is not equal to $80\% - 20\%$ as it is expected when considering a 5-folder cross-validation scheme. In the case of the benchmarks with very low degree of imbalance the training-test percentage distribution is near to $55\% - 45\%$.

We will compare the performance of FRS-OC-GA with two widely used methods in the OCC community: OCSVM[8] and SVDD [9]. The specific parameters setting for the OCC classical methods is showed in Table II.

The specific parameters setting for FRS-OC-GA 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: 0.1

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

TABLE I
SUMMARY DESCRIPTION FOR DATASETS

Dataset	#Ex.	#Atts.	Class (min., maj.)	%Class(min.; maj.)	IR
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
Haberman	306	3	(Die; Survive)	(27.42, 73.58)	2.68
Glass0123vs456	214	9	(non-window glass; remainder)	(23.83, 76.17)	3.19
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
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

TABLE II
PARAMETERS OF THE CLASSICAL OCC METHODS

Parameter	OCSVM	SVDD
kernel type	RBF	polynomial C
C	10.0	5.0
Tolerance	0.05	0.01
Epsilon	1,0E - 12	1,0E - 12
parameter optimization	quadratic programming	quadratic programming

- Parameters of the RB generation algorithm (Section III-A):
  - $threshold\_covering$: 0.07
  - $threshold\_support$: 0.05
- Parameter of the fuzzy reasoning method (Section III-B):
  - $threshold\_rule$: 0.3

In this paper, we use statistical test support to the analysis of the results [29][30]. We will use non-parametric tests, due to the fact that the initial conditions that guarantee the reliability of the parametric tests may not be satisfied [31]. Specifically, we apply the Wilcoxon signed-rank test [30] as non-parametric statistical procedure for performing pairwise comparisons between two algorithms. We compute the p-value associated to each comparison, which represents the lowest level of significance of a hypothesis that results in a rejection. Therefore, we can know whether two algorithms are significantly different or there are not significative differences between them.

Table III shows the results in performance (average of the five partitions) over the test data sets using the AUC metric, one of the most adequate metric when both classes have the same relevance and the class distribution is imbalanced as stated in Section II. For our proposal of FRBCS model (FRS-OC-GA) the average of the number of rules is also showed (#Rules) in order to measure the interpretability of the model. As the models generated by OCSVM and SVDD can not be interpreted, there are not characteristics that allow us to measure the interpretability of these models.

As can be observed in Table III, the performance of the three methods is very similar. In order to probe that similarity of behavior, Table IV presents the results of the Wilcoxon signed-rank test, showing the rankings R+ and R- values achieved and its associated p-values. As it can be observed, the p-values are very high. So, the test clearly non reject the equality hypothesis and it can be stated that there are no significant differences between FRS-OC-GA and both classical OCC approaches.

On the other hand, the number of rules of the FRBCS obtained for FRS-OC-GA is very low, thus presenting a model easy to interpret. We should note that both OCSVM and SVDD results are almost impossible to interpret and therefore they fail to deliver valuable background information about the considered one-class problem (which is of high importance, e.g., in medical domain). So, our method has achieved the proposed objective: to obtain compact OCC models with similar performance than the most used algorithms in the OCC community.

VI. CONCLUSIONS

This contribution deals with one-class classification and proposes a process for learning the whole KB of a FRBCS. The method includes a GA for the definition of the FRBCS Data Base (granularity learning, domains and non uniform fuzzy partitions), a new Rule Base generation method specific for OCC and a new fuzzy reasoning method for OCC. The fuzzy models obtained perform similar to the most commonly used algorithms in OCC and contain a reduced

TABLE III

DETAILED TABLE OF RESULTS FOR THE DIFFERENT METHODS IN THE TEST DATA SET

Dataset	OCSVM	SVDD	FRS-OC-GA	
	AUC	AUC	AUC	#Rules
Glass1	63.30	65.23	50.76	12.4
Ecoli0vs1	10.32	11.36	07.26	5.0
Wisconsin	90.11	92.01	96.49	11.2
Pima	58.73	63.71	65.04	18.0
Iris0	97.14	98.02	85.00	9.6
Glass0	50.27	48.21	41.87	13.8
Yeast1	62.64	61.46	51.08	11.8
Haberman	40.21	42.08	52.37	14.8
Glass0123vs456	62.05	62.83	82.40	6.8
Ecoli1	69.25	71.12	74.23	15.2
New-Thyroid2	81.32	78.98	86.06	3.6
New-Thyroid1	70.57	71.26	87.77	3.8
Ecoli2	64.38	62.31	72.11	15.4
Glass6	71.27	68.23	81.74	10.2
Yeast3	76.24	77.80	54.18	11.8
Ecoli3	59.11	60.36	69.76	17.16
Mean	67.77	68.24	66.13	11.36

number of rules, with a low average per system. Therefore, the obtained models are easy to interpret, as it is usual when using fuzzy rule-based models in comparison with the black box associated to a support vector machine.

TABLE IV

RESULTS OBTAINED BY THE WILCOXON TEST TO COMPARE FRS-OC-GA WITH THE CLASSICAL OCC ALGORITHMS

Comparison	R^+	R^-	P-value
FRS-OC-GA vs OCSVM	75.0	61.0	0.698152
FRS-OC-GA vs SVDD	73.0	63.0	0.776105

In future works, we must analyze in depth the associated rule based systems and to improve the precision maintaining the high interpretability level. We must check if we can use hierarchical granularity levels for managing the covering level for each class, improving the final result.

REFERENCES

[1] D. Tax, R. Duin. *Uniform object generation for optimizing one-class classifiers.* Journal of Machine Learning Research 2:155–173, 2001.

[2] D. M. J. Tax, *One-class classification*. PhD thesis, 2001.

[3] O. Cordón, F. Herrera, L. Magdalena, P. Villar. *A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base.* Information Sciences 136(1-4):85–107, 2001.

[4] P. Villar, A. Fernández, R. Carrasco, F. Herrera. *Feature Selection and Granularity Learning in Genetic Fuzzy Rule-Based Classification Systems for Highly Imbalanced Data-Sets.* International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 20(3):369-397, 2012.

[5] G. Cabral, A. Oliveira, C. Cahú. *Combining nearest neighbor data description and structural risk minimization for one-class classification.* Neural Computation & Applications 18:175–183, 2009.

[6] J. Tian, H. Gu, C. Gao, J. lian. *Local density one-class support vector machines for anomaly detection.* Nonlinear Dynamics 64:127–139, 2011.

[7] B. Krawczyk. *Diversity in Ensembles for One-Class Classification.* New Trends in Databases and Information Systems, Workshop Proceedings of the 16th East European Conference, ADBIS, pp:119-129, 2012.

[8] Y. Chen, X. S. Zhou, and T. S. Huang, *One-class svm for learning in image retrieval.* in IEEE International Conference on Image Processing, vol. 1, pp. 34-37, 2001.

[9] D. M. J. Tax and R. P. W. Duin, *Support vector data description.* Machine Learning, vol. 54(1): 45-66, 2004.

[10] M. Moya, M. Koch, L. Hostetler. *One-class classifier networks for target recognition applications.* In Proc. of World Congress on Neural Networks, pp:797-801, 1993.

[11] H. Yu, J. Han, K. Chang. *PEBL: Web page classification without negative examples.* IEEE Transactions on Knowledge and Data Engineering 16(1), 2004

[12] B. Gardner, A.M. Krieger, G. Vachtsevanos, B. Litt. *One-class novelty detection for seizure analysis from intracranial eeg.* Journal of Machine Learning Research 7: 10251044, 2006

[13] H.T. Alashwal, S. Deris, R.M. Othman. *One-class support vector machines for protein-protein interactions prediction.* International Journal Biomedical Sciences 1(2): 120127, 2006

[14] D. Sun, Q. Tran, H. Duan, G. Zhang, G. *A novel method for chinese spam detection based on one-class support vector machine.* Journal of Information and Computational Science 2(1): 109114, 2005

[15] O. Cordón, F. Herrera, 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 25(3):187-216, 2000.

[16] D. M. J. Tax, P. Juszczak, E. Pekalska, and R. P. W. Duin, *Outlier detection using ball descriptions with adjustable metric.* in Proceedings of the 2006 joint IAPR international conference on Structural, Syntactic, and Statistical Pattern Recognition, ser. SSPR'06/SPR'06. Berlin, Heidelberg: Springer-Verlag, pp. 587-595, 2006.

[17] David M. J. Tax, Klaus-Robert Muller. *A Consistency-Based Model Selection for One-Class Classification.* Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 3: 363-366, 2004.

[18] G. Cohen, H. Sax, and A. Geissbuhler, *Novelty detection using oneclass parzen density estimator. an application to surveillance of nosocomial infections.* in Studies in Health Technology and Informatics, 136:21-26, 2008.

[19] L. Manevitz and M. Yousef, *One-class document classification via neural networks.* Neurocomputing 70(7-9): 1466-1481, 2007.

[20] T. Wilk and M. Woźniak, *Soft computing methods applied to combination of one-class classifiers*. Neurocomputing 75: 185-193, 2012.

[21] B. Krawczyk and M. Woźniak, *Diversity measures for one-class classifier ensembles.* Neurocomputing 126: 36-44, 2014.

[22] 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.

[23] J.E. Baker. *Reducing bias and inefficiency in the selection algoritmh.* In Proc. second Int. Conference on Genetic Algorithms (ICGA'87):14-21, Hillsdale, 1987.

[24] F. Herrera, M. Lozano, J.L. Verdegay. *Fuzzy connectives based crossover operators to model genetic algorihtms population diversity.* Fuzzy Sets and Systems 92(1):21-30, 1997.

[25] P. Thrift. *Fuzzy logic synthesis with genetic algorithms.* Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA'91), pp:509-513, 1991.

[26] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, 1996.

[27] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. *KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework.* Journal of Multi-Valued Logic and Soft Computing 17(2-3): 255-287, 2011.

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

[29] 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.

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

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