Knowledge-Based Systems 25 (2012) 3–12

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

# Evolutionary-based selection of generalized instances for imbalanced classification

Salvador García [a,*], Joaquín Derrac [b], Isaac Triguero [b], Cristóbal J. Carmona [a], Francisco Herrera [b]

[a] University of Jaén, Department of Computer Science, 23071 Jaén, Spain
[b] University of Granada, Department of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), 18071 Granada, Spain

ARTICLE INFO

*Article history:*
Available online 4 February 2011

*Keywords:*
Nested generalized exemplar learning
Imbalanced classification
Evolutionary algorithms
Instance selection
Rule induction

ABSTRACT

In supervised classification, we often encounter many real world problems in which the data do not have an equitable distribution among the different classes of the problem. In such cases, we are dealing with the so-called imbalanced data sets. One of the most used techniques to deal with this problem consists of preprocessing the data previously to the learning process. This paper proposes a method belonging to the family of the nested generalized exemplar that accomplishes learning by storing objects in Euclidean *n*-space. Classification of new data is performed by computing their distance to the nearest generalized exemplar. The method is optimized by the selection of the most suitable generalized exemplars based on evolutionary algorithms. An experimental analysis is carried out over a wide range of highly imbalanced data sets and uses the statistical tests suggested in the specialized literature. The results obtained show that our evolutionary proposal outperforms other classic and recent models in accuracy and requires to store a lower number of generalized examples.

## 1. Introduction

The class imbalance classification problem is one of the current challenges in data mining [49]. It appears when the number of instances of one class is much lower than the instances of the other class(es) [11]. Since standard learning algorithms are developed to minimize the global measure of error, which is independent of the class distribution, in this context this causes a bias towards the majority class in the training of classifiers and results in a lower sensitivity in detecting the minority class examples. Imbalance in class distribution is pervasive in a variety of real-world applications, including but not limited to telecommunications, WWW, finance, biology and medicine [39].

A large number of approaches have been previously proposed to deal with this problem [32], which can be mainly categorized into two groups: the internal approaches which create new algorithms or modify existing ones to take the class imbalance problem into consideration [41,30,19] and external approaches which preprocess the data in order to diminish the effect caused by their class imbalance [5,10]. Imbalanced classification is also very related to cost-sensitive classification [10,50].

Exemplar-based learning was originally proposed by Medin and Schaffer [40] and revisited by Aha et al. [1] and considers a set of methods widely used in machine learning and data mining [34]. A similar scheme for learning from examples is based on the Nested Generalized Exemplar (NGE) theory. It was introduced by Salzberg [43] and makes several significant modifications to the exemplar-based learning model. The most important one is that it retains the notion of storing verbatim examples in memory but also allows examples to be generalized. They are strongly related to the nearest neighbor classifier (NN) [13] and were proposed in order to extend it.

In NGE theory, generalizations take the form of hyperrectangles in an Euclidean *n*-space [35]. Several works argue the benefits of using generalized instances together with instances to form the classification rule [48,16,38]. With respect to instance-based classification [1], the use of generalizations increases the comprehension of the data stored to perform classification of unseen data and the achievement of a substantial compression of the data, reducing the storage requirements. Considering rule induction [23,20], the ability of modeling decision surfaces by hybridizations between distance-based methods (Voronoi diagrams) and parallel axis separators could improve the performance of the models in domains with clusters of exemplars or exemplars strung out along a curve. In addition, NGE learning allows to capture generalizations with exceptions.

A main process in data mining is the one known as data reduction [42]. In classification, it aims to reduce the size of the training set mainly to increase the efficiency of the training phase (by removing redundant data) and even to reduce the classification error rate (by removing noisy data). Instance Selection (IS) is

---

* Corresponding author.
*E-mail addresses:* sglopez@ujaen.es (S. García), jderrac@decsai.ugr.es (J. Derrac), triguero@decsai.ugr.es (I. Triguero), ccarmona@ujaen.es (C.J. Carmona), herrera@decsai.ugr.es (F. Herrera).

0950-7051/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2011.01.012

one of the most known data reduction techniques in data mining [37].

The problem of yielding an optimal number of generalized examples for classifying a set of points is NP-hard. A large but finite subset of them can be easily obtained following a simple heuristic algorithm acting over the training data. However, almost all generalized examples produced could be irrelevant and, as a result, the most influential ones must be distinguished. Evolutionary Algorithms (EAs) [17] have been used in data mining with promising results [22]. They have been successfully used for descriptive [8] and predictive tasks [2], nearest neighbor classification [47,46], feature selection [31,44,36], IS [7,24], simultaneous instance and feature selection [15] and under-sampling for imbalanced learning [25,29]. NGE is also directly related to clustering and EAs have been extensively used for this problem [33].

In this paper, we propose the use of EAs for generalized instances selection in imbalanced classification domains. Our objective is to increase the accuracy of this type of representation by means of selecting the best suitable set of generalized examples to enhance its classification performance over imbalanced domains. We compare our approach with the most representative models of NGE learning: BNGE [48], RISE [16] and INNER [38], and two well-known rule induction learning methods: RIPPER [12] and PART [21].

We have selected a large collection of imbalanced data sets from KEEL-dataset repository[1] [3] for developing our experimental analysis. In order to deal with the problem of imbalanced data sets we will include an study that involves the use of a preprocessing technique, the “Synthetic Minority Over-sampling Technique” (SMOTE) [9], to balance the distribution of training examples in both classes. The empirical study has been checked via non-parametrical statistical testing [14,28,27], and the results show an improvement of accuracy for our approach whereas the number of generalized examples stored in the final subset is much lower.

The rest of this paper is organized as follows: Section 2 gives an explanation of NGE learning. In Section 3, we introduce some issues of imbalanced classification: the SMOTE pre-processing technique and the evaluation metric used for this scenario. Section 4 explains all topics concerning the approach proposed to tackle the imbalanced classification problem. Sections 5 and 6 describe the experimental framework used and the analysis of results, respectively. Finally, in Section 7, we point out the conclusions achieved.

## 2. NGE learning

NGE is a learning paradigm based on class exemplars, where an induced hypothesis has the graphical shape of a set of hyperrectangles in an $M$-dimensional Euclidean space. Exemplars of classes are either hyperrectangles or single instances [43]. The input of an NGE system is a set of training examples, each described as a vector of pairs *numeric_attribute/value* and an associated class. Attributes can either be numerical or categorical. Numerical attributes are usually normalized in the [0, 1] interval.

In NGE, an initial set of points given in the $M$-dimensional Euclidean space set is generalized into a smaller set of hyperrectangles in terms of the elements that it contains. Choosing which hyperrectangle is generalized from a subset of points or other hyperrectangles and how it is generalized depends on the concrete NGE algorithm employed.

In the next subsections we describe the essential concepts to understand the NGE learning model, as well as the algorithms used in this study. First, we explain the necessary concepts to understand the classification rule followed by this type of method (Section 2.1). After this, in Section 2.2, the two classical proposals of hyperrectangle learning will be briefly described, BNGE in Section 2.2.1 and RISE in Section 2.2.2, followed by the advanced approach INNER in Section 2.2.3.

### 2.1. Matching and classification

The matching process is one of the central features in NGE learning and it allows some customization, if desired. Generally speaking, this process computes the distance between a new example and an exemplar memory object (a generalized example). For the remainder of this paper, we will refer to the example to be classified as $E$ and the generalized example as $G$, independently of whether $G$ is formed by a single point or it has some volume.

The model computes a match score between $E$ and $G$ by measuring the Euclidean distance between two objects. The Euclidean distance is well-known when $G$ is a single point. Otherwise, the distance is computed as follows (considering numerical attributes):

$$D_{EG} = \sqrt{\sum_{i=1}^{M} \left( \frac{dif_i}{max_i - min_i} \right)^2},$$

$$dif_i = \begin{cases} E_{f_i} - G_{upper} \text{ when } E_{f_i} > G_{upper}, \\ G_{lower} - E_{f_i} \text{ when } E_{f_i} < G_{lower}, \\ 0 \text{ otherwise}, \end{cases}$$

where $M$ is the number of attributes of the data, $E_{f_i}$ is the value of the $i$th attribute of the example, $G_{upper}$ and $G_{lower}$ are the upper and lower values of $G$ for a specific attribute and $max_i$ and $min_i$ are the maximum and minimum values for $i$th attribute in training data, respectively.

The distance measure represents the length of a line dropped perpendicularly from the point $E_{f_i}$ to the nearest surface, edge or corner of $G$. Note that internal points to a hyperrectangle have distance 0 to that rectangle. In the case of overlapping rectangles, several strategies could be followed, but usually is accepted a point falling in the overlapping area belongs to the smaller rectangle (the size of a hyperrectangle is defined in terms of volume). The volume is computed following the indications given in [48]. In nominal attributes, the distance is 0 when two attributes have the same categorical label, and 1 on the contrary.

### 2.2. Proposals for NGE learning

EACH, BNGE and RISE are the pioneer proposals for NGE learning. EACH is not considered in this paper because the authors of BNGE demonstrated that their proposal clearly outperforms EACH. INNER is a more recent approach also introduced in this section.

#### 2.2.1. BNGE: Batch nested generalized exemplar

BNGE is a batch version of the first model of NGE (also known as EACH [43]) and it is proposed to alleviate some drawbacks presented in it [48]. The generalization of examples is done by expanding their boundaries just to cover the desired example merging generalized instances only if the new generalized example does not cover (or overlap with) any stored example from a different class. It does not permit overlapping or nesting.

#### 2.2.2. RISE: Unifying instance-based and rule-based induction

RISE [16] is an approach proposed to overcome some of the limitations of instance-based learning and rule induction by unifying the two. It follows similar guidelines explained above, but it furthermore introduces some improvements regarding distance computations and selection of the best rule using the Laplace

---

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

correction, present in many existing rule-induction techniques [23].

### 2.2.3. INNER: Inflating examples to obtain rules

INNER [38] starts by selecting a small random subset of examples, which are iteratively inflated in order to cover the surroundings with examples of the same class. Then, it applies a set of elastic transformations over the rules, to finally obtain a concise and accurate rule set to classify.

## 3. Imbalanced data sets in classification

In this section, we address some important issues related to imbalanced classification by describing the pre-processing technique applied to deal with the imbalance problem: the SMOTE algorithm [9]. Also, we will present the evaluation metric mainly used for this type of classification problem.

### 3.1. Pre-processing imbalanced data sets. The SMOTE algorithm

As mentioned before, applying a pre-processing step in order to balance the class distribution is a suitable solution to the imbalanced data set problem [5]. Specifically, in this work we have chosen an over-sampling method which is widely used in this area: the SMOTE algorithm [9].

In this approach, the positive class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the $k$ minority class nearest neighbors. This process is illustrated in Fig. 1, where $x_i$ is the selected point, $x_{i1}$ to $x_{i4}$ are some selected nearest neighbors and $r_1$ to $r_4$ are the synthetic data points created by the randomized interpolation procedure.

Synthetic samples are generated in the following way: Take the difference between the feature vector (sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features. This approach effectively forces the decision region of the minority class to become more general.

### 3.2. Evaluation in imbalanced domains

One appropriate metric that could be used to measure the performance of classification over imbalanced data sets is the Receiver Operating Characteristic (ROC) graphics [6]. In these graphics, the tradeoff between the benefits and costs can be visualized, and acknowledges the fact that the capacity of any classifier cannot increase the number of true positives without also increasing the false positives. The Area Under the ROC Curve (AUC) corresponds to the probability of correctly identifying which of the two stimuli is noise and which is signal plus noise. AUC provides a single-number summary for the performance of learning algorithms.

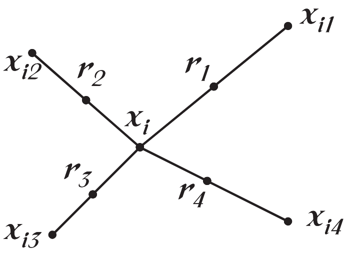

**Fig. 1.** An illustration of how to create the synthetic data points in the SMOTE algorithm.

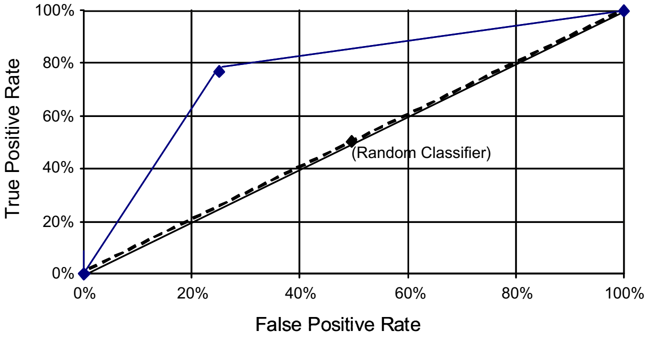

**Fig. 2.** Example of an ROC plot. Two classifiers are represented: the solid line is a good performing classifier whereas the dashed line represents a random classifier.

The way to build the ROC space is to plot on a two-dimensional chart the true positive rate (Y axis) against the false positive rate (X axis) as shown in Fig. 2. The points (0,0) and (1,1) are trivial classifiers in which the output class is always predicted as negative and positive respectively, while the point (0,1) represents perfect classification. To compute the AUC we just need to obtain the area of the graphic as:

$$AUC = \frac{1 + True\_Positive\_Rate - False\_Positive\_Rate}{2}. \tag{1}$$

## 4. Evolutionary selection of generalized examples for imbalanced classification

The approach proposed in this paper, named Evolutionary Generalized Instance Selection by CHC (EGIS-CHC), is fully explained in this section. Firstly, we introduce the CHC model used as an EA to perform generalized instance selection in Section 4.1. Secondly, the specific issues regarding representation and fitness function are specified in Section 4.2. Finally, Section 4.3 describes the process for generating the initial set of generalized examples.

### 4.1. CHC model

As an evolutionary computation method, we have used the CHC model [18]. CHC is a classical evolutionary model that introduces different features to obtain a trade-off between exploration and exploitation; such as incest prevention, reinitialization of the search process when it becomes blocked and the competition among parents and offspring into the replacement process. We have selected CHC due to the fact that it has been widely studied, being now a well-known algorithm on evolutionary computation. Furthermore, previous studies like [7,25] support the fact that it can perform well on data reduction problems.

During each generation the CHC develops the following steps:

- It uses a parent population of size $N$ to generate an intermediate population of $N$ individuals, which are randomly paired and used to generate $N$ potential offspring.
- Then, a survival competition is held where the best $N$ chromosomes from the parent and offspring populations are selected to form the next generation.

CHC also implements HUX recombination operator. HUX exchanges half of the bits that differ between parents, where the bit position to be exchanged is randomly determined. It also employs a method of incest prevention: Before applying HUX to two parents, the Hamming distance between them is measured. Only those parents who differ from each other by some number of bits (mating threshold) are mated. If no offspring is inserted into the new population then the threshold is reduced.

No mutation is applied during the recombination phase. Instead, when the search stops making progress the population is reinitialized to introduce new diversity. The chromosome representing the best solution found is used as a template to re-seed the population, randomly changing 35% of the bits in the template chromosome to form each of the other chromosomes in the population.

### 4.2. Representation and fitness function

Let us assume that there is a training set *TR* with *P* instances and each one of them has *M* input attributes. Let us also assume that there is a set of generalized instances *GS* with *N* generalized instances and each one of the *N* generalized instances has *M* conditions which can be numeric conditions, expressed in terms of minimum and maximum values in interval $[0,1]$; or they can be categorical conditions, assuming that there are $v$ different values for each attribute. Let $S \subseteq GS$ be the subset of selected generalized instances resulted in the run of a generalized instances selection algorithm.

Generalized instance selection can be considered as a search problem in which EAs can be applied. We take into account two important issues: the specification of the representation of the solutions and the definition of the fitness function.

- *Representation:* The search space associated is constituted by all the subsets of *GS*. This is accomplished by using a binary representation. A chromosome consists of *N* genes (one for each sample in *GS*) with two possible states: 0 and 1. If the gene is 1, its associated generalized example is included in the subset of *GS* represented by the chromosome. If it is 0, this does not occur.
- *Fitness Function:* Let *S* be a subset of samples of *GS* and be coded by a chromosome. We define a fitness function based on AUC evaluated over *TR* through the rule described in Section 2.1.

$$Fitness(S) = \alpha \cdot AUC + (1 - \alpha) \cdot red\_rate,$$

where *AUC* denotes the computation of the AUC measure from *TR* using *S*. *red_rate* denotes the ratio of generalized examples selected. The objective of the EAs is to maximize the fitness function defined. We preserve the value of $\alpha$ = 0.5 used in previous works related to IS [7].

In order to perform the classification of an unseen example, our approach uses the mechanism described in [43]. In short, they are:

- If no hyperrectangle covers the example, the class of the nearest hyperrectangle defines the prediction.
- If various hyperrectangles cover the example, the one with lowest volume is the chosen to predict the class, allowing exceptions within generalizations.

### 4.3. Obtention of the initial set of hyperrectangles

In our approach, the initial set of hyperrectangles is computed using a simple and fast heuristic which obtains good results. It yields a generalization from each example in the training set, finding, or each one, the $K - 1$ nearest neighbors being the *K*th

**Table 1**
Description for imbalanced data sets.

| Data sets | #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 |
| vehicle2 | 846 | 18 | (bus, remainder) | (28.37,71.63) | 2.52 |
| vehicle3 | 846 | 18 | (opel, remainder) | (28.37,71.63) | 2.52 |
| haberman | 306 | 3 | (die, survive) | (27.42,73.58) | 2.68 |
| glass0123vs456 | 214 | 9 | (non-window glass, remainder) | (23.83,76.17) | 3.19 |
| vehicle0 | 846 | 18 | (van, remainder) | (23.64,76.36) | 3.23 |
| ecoli1 | 336 | 7 | (im, remainder) | (22.92,77.08) | 3.36 |
| new-thyroid2 | 215 | 5 | (hypo, remainder) | (16.89,83.11) | 4.92 |
| new-thyroid1 | 215 | 5 | (hyper, remainder) | (16.28,83.72) | 5.14 |
| ecoli2 | 336 | 7 | (pp, remainder) | (15.48,84.52) | 5.46 |
| segment0 | 2308 | 19 | (brickface, remainder) | (14.26,85.74) | 6.01 |
| glass6 | 214 | 9 | (headlamps, remainder) | (13.55,86.45) | 6.38 |
| yeast3 | 1484 | 8 | (me3, remainder) | (10.98,89.02) | 8.11 |
| ecoli3 | 336 | 7 | (imU, remainder) | (10.88,89.12) | 8.19 |
| 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 |
| 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 |
| yeast4 | 1484 | 8 | (me2, remainder) | (3.43,96.57) | 28.41 |
| 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 |

neighbor an example of a different class. Then, each example is expanded considering these $K-1$ neighbors by using, in the case of numerical attributes, the minimal and maximal values as the limits of the interval defined, or getting all the different categorical values, in the case of nominal attributes, to form a subset of possible values from them.

Once all the generalized examples are obtained, the duplicated ones are removed. Hence $|GS| \leqslant |TR|$.

## 5. Experimental framework

This section describes the methodology followed in the experimental study of the generalized examples based learning approaches. We will explain the configuration of the experiment: imbalanced data sets used and parameters for the algorithms.

**Table 2**
Parameters considered for the algorithms.

| Algorithm | Parameters |
|---|---|
| EGIS-CHC | *Pop.Size* = 50, *Num.Eval.* = 10000, $\alpha = 0.5$ |
| BNGE | It has not parameters to be fixed |
| RISE | $Q = 1$, $S = 2$ |
| INNER | Initial Instances = 10, MaxCycles = 5, Min Coverage = 0.95, Min Presentations = 3000, Iterations to Regularize = 50, Select Threshold = −50.0 |
| RIPPER | Grow_pct = 0.66, K = 2 |
| PART | Level of Conficende = 0.25, Items per Leaf = 2 |

### 5.1. Data sets and parameters

In this study, EGIS-CHC is applied to thirty-six binary data sets from the KEEL-dataset repository [3] with different imbalance ratios (IR) [41]. Table 1 summarises the data selected in this study and shows, for each data set, 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 ordered by the IR, from low to high imbalanced data sets.

The data sets considered are partitioned using the *five fold cross-validation (5-fcv)* procedure. The parameters of the used algorithms are presented in Table 2. All the methods were run using KEEL software [4], following the recommended parameter values given in the KEEL platform to configure the methods, which also correspond to the settings used in the bibliography of these methods. Stochastic methods have been run three times with different random number seeds.

Regarding the use of the SMOTE pre-processing method [9], we consider only the 1-nearest neighbor (using the euclidean distance) to generate the synthetic samples, and we balance both classes to the 50% distribution.

### 5.2. Statistical tests for evaluation

In this paper, we use the hypothesis testing techniques to provide statistical support to the analysis of the results [26]. Specifically, we will use non-parametric tests due to the fact that the initial conditions that guarantee the reliability of the parametric

**Table 3**
*AUC* in test data resulted from the run of the approaches over the data sets considered.

| | EGIS-CHC | | 1NN | | BNGE | | RISE | | INNER | | RIPPER | | PART | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | SD | AUC | SD | AUC | SD | AUC | SD | AUC | SD | AUC | SD | AUC | SD |
| glass1 | 0.7521 | 0.0530 | **0.7888** | 0.0253 | 0.6956 | 0.0165 | 0.6899 | 0.0785 | 0.6657 | 0.0205 | 0.7396 | 0.0858 | 0.6651 | 0.0369 |
| ecoli0vs1 | 0.9667 | 0.0263 | 0.9626 | 0.0186 | 0.9764 | 0.0138 | 0.9205 | 0.0425 | **0.9867** | 0.0112 | 0.9582 | 0.0149 | 0.9832 | 0.0238 |
| wisconsin | 0.9608 | 0.0104 | 0.9551 | 0.0115 | **0.9685** | 0.0099 | 0.9289 | 0.0146 | 0.9449 | 0.0109 | 0.9638 | 0.0099 | 0.9518 | 0.0276 |
| pima | 0.7278 | 0.0312 | 0.6643 | 0.0180 | 0.7005 | 0.0188 | 0.6583 | 0.0416 | 0.6081 | 0.0161 | 0.6966 | 0.0467 | 0.7118 | 0.0055 |
| iris0 | **1.0000** | 0.0000 | **1.0000** | 0.0000 | **1.0000** | 0.0000 | 0.9900 | 0.0224 | **1.0000** | 0.0000 | 0.9789 | 0.0472 | 0.9900 | 0.0224 |
| glass0 | 0.7404 | 0.0547 | **0.8270** | 0.0226 | 0.7403 | 0.0188 | 0.7754 | 0.0528 | 0.5581 | 0.0211 | 0.7680 | 0.0637 | 0.8161 | 0.0601 |
| yeast1 | **0.6929** | 0.0248 | 0.6340 | 0.0137 | 0.6522 | 0.0113 | 0.6134 | 0.0341 | 0.6110 | 0.0149 | 0.6772 | 0.0259 | 0.6825 | 0.0339 |
| haberman | 0.5882 | 0.0741 | 0.5473 | 0.0185 | 0.5244 | 0.0259 | 0.5397 | 0.0722 | **0.5920** | 0.0202 | 0.5554 | 0.0373 | 0.5588 | 0.0479 |
| vehicle2 | 0.9219 | 0.0333 | 0.9404 | 0.0141 | 0.8183 | 0.0228 | 0.9168 | 0.0457 | 0.5613 | 0.0190 | **0.9533** | 0.0253 | 0.9521 | 0.0063 |
| vehicle3 | **0.7298** | 0.0387 | 0.6695 | 0.0180 | 0.5751 | 0.0162 | 0.6578 | 0.0258 | 0.5000 | 0.0054 | 0.7101 | 0.0508 | 0.5934 | 0.0342 |
| glass0123vs456 | 0.9210 | 0.0255 | 0.9124 | 0.0114 | 0.8834 | 0.0211 | 0.9130 | 0.0488 | 0.8660 | 0.0139 | 0.9053 | 0.0490 | 0.8757 | 0.0573 |
| vehicle0 | 0.9145 | 0.0131 | 0.9112 | 0.0118 | 0.6156 | 0.0194 | 0.8235 | 0.0265 | 0.5000 | 0.0051 | 0.9088 | 0.0435 | **0.9348** | 0.0268 |
| ecoli1 | 0.8865 | 0.0358 | 0.7969 | 0.0214 | 0.7904 | 0.0176 | 0.8500 | 0.0684 | 0.8477 | 0.0204 | **0.9152** | 0.0404 | 0.8536 | 0.0546 |
| new-thyroid1 | **0.9944** | 0.0076 | 0.9774 | 0.0102 | 0.9857 | 0.0102 | 0.9571 | 0.0391 | 0.9087 | 0.0163 | 0.9288 | 0.0641 | 0.9317 | 0.0697 |
| new-thyroid2 | **0.9861** | 0.0170 | 0.9802 | 0.0113 | 0.9258 | 0.0139 | 0.9286 | 0.0714 | 0.9179 | 0.0163 | 0.9275 | 0.0340 | 0.9345 | 0.0384 |
| ecoli2 | 0.8618 | 0.1152 | **0.9062** | 0.0195 | 0.8429 | 0.0156 | 0.8566 | 0.0385 | 0.8616 | 0.0217 | 0.8618 | 0.0701 | 0.8288 | 0.0710 |
| segment0 | 0.9851 | 0.0083 | **0.9952** | 0.0060 | 0.9453 | 0.0073 | 0.9500 | 0.0276 | 0.8272 | 0.0191 | 0.9882 | 0.0108 | 0.9848 | 0.0064 |
| glass6 | **0.8803** | 0.1144 | 0.8713 | 0.0161 | 0.8546 | 0.0022 | 0.8546 | 0.0994 | 0.7746 | 0.0222 | 0.8449 | 0.0989 | 0.8505 | 0.0877 |
| yeast3 | 0.8890 | 0.0193 | 0.8181 | 0.0081 | 0.7616 | 0.0119 | 0.7521 | 0.0590 | 0.8278 | 0.0141 | **0.9181** | 0.0268 | 0.8169 | 0.0918 |
| ecoli3 | **0.8436** | 0.0637 | 0.7448 | 0.0131 | 0.7295 | 0.0129 | 0.6867 | 0.0609 | 0.6438 | 0.0204 | 0.8069 | 0.0990 | 0.6631 | 0.0689 |
| yeast2vs4 | 0.8609 | 0.0705 | 0.8521 | 0.0133 | 0.7824 | 0.0103 | 0.8657 | 0.0392 | 0.6710 | 0.0093 | **0.8947** | 0.0601 | 0.8139 | 0.0732 |
| yeast05679vs4 | 0.7558 | 0.0445 | 0.7023 | 0.0125 | 0.6159 | 0.0103 | 0.6693 | 0.0800 | 0.6986 | 0.0134 | **0.7828** | 0.0741 | 0.6678 | 0.0714 |
| vowel0 | 0.9489 | 0.0659 | **1.0000** | 0.0000 | 0.8378 | 0.0124 | 0.9472 | 0.0483 | 0.6444 | 0.0148 | 0.9573 | 0.0637 | 0.9244 | 0.0555 |
| glass016vs2 | 0.6593 | 0.1588 | 0.5767 | 0.0288 | 0.5000 | 0.0114 | 0.6631 | 0.1227 | 0.5000 | 0.0114 | 0.6407 | 0.1325 | **0.6767** | 0.1088 |
| glass2 | 0.6242 | 0.1205 | 0.6008 | 0.0244 | 0.5000 | 0.0112 | 0.5437 | 0.1053 | 0.5650 | 0.0331 | 0.3666 | 0.3551 | **0.6611** | 0.1762 |
| shuttle0vs4 | 0.9960 | 0.0089 | 0.9960 | 0.0035 | 0.9960 | 0.0035 | 0.9950 | 0.0038 | 0.9880 | 0.0061 | **0.9997** | 0.0007 | **0.9997** | 0.0007 |
| yeast1vs7 | 0.6895 | 0.0928 | 0.6457 | 0.0161 | 0.5775 | 0.0142 | **0.6898** | 0.0827 | 0.5143 | 0.0069 | 0.6770 | 0.0965 | 0.5895 | 0.0694 |
| ecoli4 | **0.9374** | 0.0684 | 0.8702 | 0.0091 | 0.8000 | 0.0141 | 0.7734 | 0.1051 | 0.7250 | 0.0139 | 0.8841 | 0.0560 | 0.8405 | 0.0924 |
| abalone9vs18 | 0.7235 | 0.0930 | 0.6037 | 0.0101 | 0.6096 | 0.0158 | **0.7307** | 0.1555 | 0.5763 | 0.0136 | 0.5337 | 0.1339 | 0.5421 | 0.0237 |
| glass016vs5 | 0.7300 | 0.2433 | 0.8357 | 0.0176 | 0.6471 | 0.0176 | 0.8471 | 0.2292 | 0.5000 | 0.0108 | 0.8414 | 0.1384 | **0.8943** | 0.2205 |
| shuttle2vs4 | **0.9500** | 0.1118 | **0.9500** | 0.0131 | **0.9500** | 0.0131 | **0.9500** | 0.1118 | **0.9500** | 0.0131 | 0.8500 | 0.2236 | **0.9500** | 0.1118 |
| yeast1458vs7 | 0.6169 | 0.0903 | 0.5735 | 0.0099 | 0.5000 | 0.0013 | 0.5053 | 0.0363 | 0.5000 | 0.0013 | **0.6744** | 0.1245 | 0.5000 | 0.0000 |
| yeast4 | **0.8018** | 0.0604 | 0.6671 | 0.0104 | 0.5280 | 0.0071 | 0.6426 | 0.0508 | 0.5260 | 0.0112 | 0.6382 | 0.0991 | 0.6412 | 0.0858 |
| ecoli0137vs26 | 0.8445 | 0.2216 | 0.8427 | 0.0122 | 0.7500 | 0.0099 | 0.7482 | 0.2500 | 0.7482 | 0.0111 | 0.8372 | 0.2181 | **0.8481** | 0.2247 |
| yeast6 | 0.8566 | 0.1034 | 0.7482 | 0.0094 | 0.5986 | 0.0067 | 0.6847 | 0.1080 | 0.5656 | 0.0142 | 0.7999 | 0.1747 | 0.7809 | 0.1323 |
| abalone19 | **0.5268** | 0.1232 | 0.4963 | 0.0057 | 0.4999 | 0.0026 | 0.4928 | 0.0047 | 0.5000 | 0.0026 | 0.2320 | 0.2122 | 0.5000 | 0.0000 |
| average | **0.8268** | 0.0679 | 0.8018 | 0.0135 | 0.7411 | 0.0124 | 0.7781 | 0.0695 | 0.6993 | 0.0138 | 0.7949 | 0.0863 | 0.7892 | 0.0644 |

tests may not be satisfied, causing the statistical analysis to lose credibility with these parametric tests [45].

We will use the Wilcoxon signed-rank test [14] as a non-parametric statistical procedure for performing pairwise comparisons between two algorithms. For multiple comparisons we use the Friedman test [26,27] to detect statistical differences among a group of results, and the Holm post hoc test [27] in order to find which algorithms are distinctive among a 1 × n comparison.

The post hoc procedure allows us to know whether a hypothesis of comparison of means could be rejected at a specified level of significance $\alpha$. However, it is very interesting to compute the $p$-value associated to each comparison, which represents the lowest level of significance of a hypothesis that results in a rejection. It is the adjusted $p$-value (APV). Furthermore, we consider the average ranking of the algorithms in order to measure how good a method is with respect to its partners. This ranking is obtained by assigning a position to each algorithm depending on its performance for each data set. The algorithm which achieves the best accuracy on a specific data set will have the first ranking (value 1); then, the algorithm with the second best accuracy is assigned rank 2, and so forth. This task is carried out for all data sets and finally an average ranking is computed as the mean value of all rankings.

These tests are suggested in the studies presented in [14,28,26,27], where their use in the field of machine learning is highly recommended. For a wider description of the use of these tests, please refer to the Website http://sci2s.ugr.es/sicidm/.

**Table 4**
Average Friedman rankings of and APVs using Holm's procedure in AUC: data sets without preprocessing.

| Algorithm | AUC ranking | AUC APV |
|---|---|---|
| EGIS-CHC | 2.208 | – |
| 1NN | 3.278 | 0.03570 |
| RIPPER | 3.528 | 0.01912 |
| PART | 3.779 | 0.00616 |
| RISE | 4.472 | 0.00003 |
| BNGE | 5.000 | $2.0941 \cdot 10^{-7}$ |
| INNER | 5.736 | $2.5534 \cdot 10^{-11}$ |

## 6. Results and analysis

In this section we will carry out a complete experimental analysis in order to show three important issues:

- First, the performance of the algorithms when they are applied over the original data sets (Section 6.1).
- Secondly, the comparison of using or not SMOTE previous to EGIS-CHC and the performance of the algorithms when they are applied over SMOTE-processed data sets (Section 6.2).
- Then, the analysis of complexity of the models obtained by means of the computation of the number of generalized instances or rules produced by each algorithm (Section 6.3).

Finally, Section 6.4 will present a summary of the analysis of results.

### *6.1. Performance analysis without pre-processing (original data sets)*

Following, we analyze the performance of the methods considering all the original, without pre-processing, data sets. The complete table of results for all the algorithms used in this study is shown in Table 3, where the reader can observe the full test results, with their associated standard deviation (SD), in order to compare the performance of each approach. The best case in each data set is highlighted in bold. We must emphasize the good results achieved by EGIS-CHC, as it obtains the highest *AUC* value among all algorithms, but we are aware that the average AUC can be very biased and next we will analyze these results by nonparametric statistical tests.

In order to analyze these results, Table 4 shows the average ranking computed for all approaches according to the *AUC* metric and the APV computed by the Holm test [27] reported between

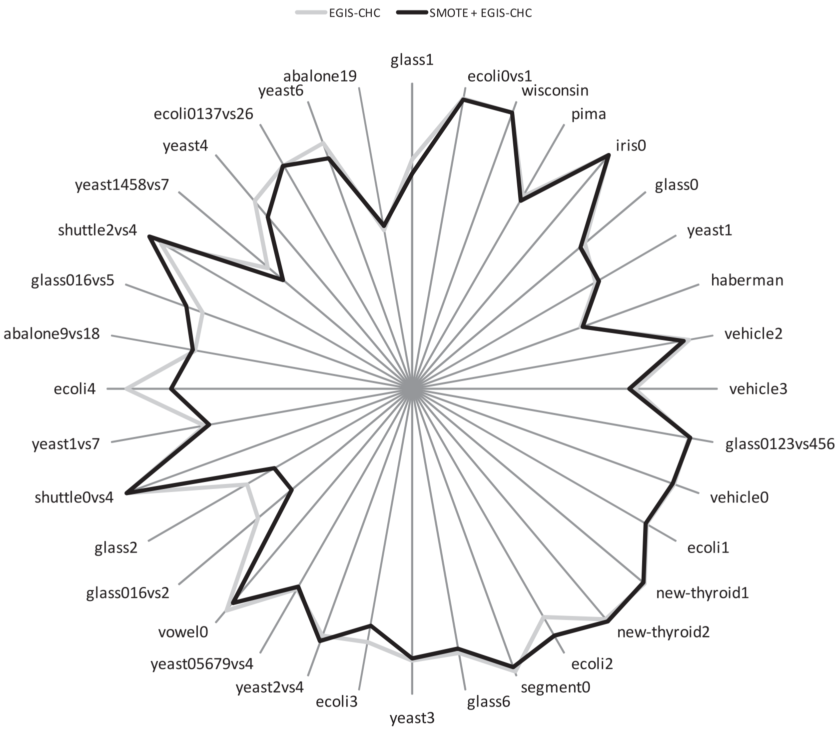

**Fig. 3.** Comparison of results between EGIS-CHC with and without SMOTE preprocessing.

our approach and the algorithm included in each row. We can observe that EGIS-CHC has obtained the lowest value in the ranking and therefore it is the best algorithm. Moreover, since all APVs are lower than a standard used level of significance of $\alpha = 0.05$, the null-hypothesis of equality is rejected in all cases, which supports the conclusion that EGIS-CHC outperforms the remaining algorithms when the original data sets are used. Note that EGIS-CHC is the only method that specifically optimized AUC instead of global measure of accuracy. Next, we will study the effects of SMOTE as preprocessing step of all the methods considered.

### *6.2. Analysis of the significance of the pre-processing mechanism: Use of SMOTE*

In this part of the experimental study, we use SMOTE to preprocess the data sets used in this paper to obtain balanced distribution of classes. In imbalanced classification, SMOTE has proved to be an excellent preprocessing step suitable to improve almost ant learning algorithm and it is considered a standard in the topic.

First of all, we focus on checking whether or not our approach improves its behavior when SMOTE is previously used. Fig. 3 plots a star graphic that represents the *AUC* metric obtained for each data set and allows us to see in a easier way how both algorithms behave in the same domains. We can observe that the use of SMOTE previous to EGIS-CHC affects negatively to the performance in most of data sets. In fact, if we conduct the Wilcoxon test using the results obtained by the two algorithms, the test informs us that EGIS-CHC without SMOTE outperforms EGIS-CHC with SMOTE with a $p$-value of 0.03, thus the use of SMOTE in EGIS-CHC is not recommendable.

Two main reasons support the conclusion achieved before. The first one is the increase of instances in training data, which produces a subsequent increase in the number of genes codified by a chromosome. It is well-known that the size of chromosome influences the performance and the tradeoff between exploration–exploitation capabilities of an EA. The second reason is the addition of noisy instances by the interpolation mechanism of SMOTE. SMOTE generates artificial data within decision boundaries and it could cause problems to the initialization process of generalized instances used by EGIS-CHC (producing irrelevant generalized instances and increasing the number of them).

In this case, the complete table of results with the application of the SMOTE pre-processing technique, except in our approach, is shown in Table 5, which follows the same structure than the previous one. Also in this case the EGIS-CHC approach again achieves the highest result in test among all the algorithms compared in this analysis.

In order to analyze these results, Table 6 shows the average ranking and APVs computed for all approaches following the same scheme of Table 4. We can observe that EGIS-CHC obtains the lowest value in the ranking and therefore it is the best algorithm. But in this case, the pairwise comparisons conducted by the Holm test indicate us that no differences between the methods compared can be detected, except for the RISE algorithm. Hence, EGIS-CHC has a similar performance using *AUC* metric than most of the other algorithms combined with SMOTE.

**Table 5**
*AUC* in test data resulted from the run of the approaches over the data sets considered preprocessed by SMOTE (except EGIS-CHC).

| | EGIS-CHC | | SMOTE + 1NN | | SMOTE + BNGE | | SMOTE + RISE | | SMOTE + INNER | | SMOTE + RIPPER | | SMOTE + PART | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | SD | AUC | SD | AUC | SD | AUC | SD | AUC | SD | AUC | SD | AUC | SD |
| glass1 | 0.7521 | 0.0530 | **0.7738** | 0.0561 | 0.6632 | 0.0968 | 0.7042 | 0.0764 | 0.7415 | 0.1079 | 0.7326 | 0.0927 | 0.6927 | 0.0622 |
| ecoli0vs1 | 0.9667 | 0.0263 | 0.9626 | 0.0302 | 0.9729 | 0.0281 | 0.9344 | 0.0384 | 0.9766 | 0.0368 | **0.9832** | 0.0238 | 0.9694 | 0.0305 |
| wisconsin | 0.9608 | 0.0104 | 0.9624 | 0.0177 | **0.9727** | 0.0106 | 0.9475 | 0.0311 | 0.9397 | 0.0278 | 0.9636 | 0.0155 | 0.9584 | 0.0087 |
| pima | 0.7278 | 0.0312 | 0.6808 | 0.0505 | 0.7186 | 0.0689 | 0.6552 | 0.0278 | 0.7064 | 0.0556 | 0.7010 | 0.0546 | **0.7312** | 0.0311 |
| iris0 | **1.0000** | 0.0000 | **1.0000** | 0.0000 | **1.0000** | 0.0000 | **1.0000** | 0.0000 | **1.0000** | 0.0000 | 0.9800 | 0.0447 | 0.9900 | 0.0224 |
| glass0 | 0.7404 | 0.0547 | **0.8595** | 0.0598 | 0.7554 | 0.0925 | 0.7504 | 0.1039 | 0.7687 | 0.0415 | 0.7905 | 0.0738 | 0.7250 | 0.1205 |
| yeast1 | **0.6929** | 0.0248 | 0.6533 | 0.0318 | 0.7032 | 0.0314 | 0.6072 | 0.0353 | 0.6793 | 0.0300 | 0.6967 | 0.0269 | 0.7049 | 0.0436 |
| haberman | 0.5882 | 0.0741 | 0.5394 | 0.0525 | 0.6071 | 0.0621 | 0.5077 | 0.0915 | **0.6162** | 0.0422 | 0.5641 | 0.0619 | 0.6086 | 0.0799 |
| vehicle2 | 0.9219 | 0.0333 | 0.9539 | 0.0202 | 0.9358 | 0.0125 | 0.8715 | 0.0755 | 0.8408 | 0.0649 | 0.9469 | 0.0288 | **0.9628** | 0.0144 |
| vehicle3 | 0.7298 | 0.0387 | 0.6835 | 0.0235 | 0.6919 | 0.0641 | 0.6082 | 0.0350 | 0.6388 | 0.0347 | 0.7198 | 0.0347 | **0.7519** | 0.0360 |
| glass0123vs456 | 0.9210 | 0.0255 | 0.9224 | 0.0154 | **0.9354** | 0.0337 | 0.9076 | 0.0424 | 0.9159 | 0.0258 | 0.8969 | 0.0524 | 0.9104 | 0.0430 |
| vehicle0 | 0.9145 | 0.0131 | 0.9106 | 0.0191 | 0.8843 | 0.0296 | 0.8075 | 0.0532 | 0.8014 | 0.0375 | 0.9351 | 0.0395 | **0.9382** | 0.0254 |
| ecoli1 | 0.8865 | 0.0358 | 0.8298 | 0.0783 | 0.8398 | 0.0476 | 0.8482 | 0.0554 | **0.8968** | 0.0226 | 0.8607 | 0.0427 | 0.8923 | 0.0266 |
| new-thyroid1 | **0.9944** | 0.0076 | 0.9774 | 0.0279 | 0.9944 | 0.0076 | **0.9944** | 0.0124 | 0.9464 | 0.0419 | 0.9520 | 0.0488 | 0.9659 | 0.0612 |
| new-thyroid2 | **0.9861** | 0.0170 | 0.9774 | 0.0279 | 0.9829 | 0.0310 | 0.9571 | 0.0639 | 0.9520 | 0.0383 | 0.9718 | 0.0413 | 0.9516 | 0.0446 |
| ecoli2 | 0.8618 | 0.1152 | 0.9343 | 0.0505 | 0.8870 | 0.0291 | 0.8496 | 0.0874 | 0.8727 | 0.0385 | 0.8603 | 0.0469 | 0.8533 | 0.1051 |
| segment0 | 0.9851 | 0.0083 | **0.9949** | 0.0066 | 0.9828 | 0.0209 | 0.9469 | 0.0338 | 0.9815 | 0.0100 | 0.9866 | 0.0083 | 0.9911 | 0.0073 |
| glass6 | 0.8803 | 0.1144 | 0.8686 | 0.0867 | 0.8659 | 0.0587 | 0.8244 | 0.0327 | 0.8923 | 0.0600 | 0.8869 | 0.1097 | **0.9090** | 0.0547 |
| yeast3 | 0.8890 | 0.0193 | 0.8607 | 0.0134 | 0.8824 | 0.0294 | 0.7140 | 0.0476 | **0.9122** | 0.0178 | 0.8911 | 0.0083 | 0.8966 | 0.0349 |
| ecoli3 | 0.8436 | 0.0637 | 0.7777 | 0.0482 | 0.8491 | 0.0908 | 0.5851 | 0.0638 | 0.8603 | 0.0746 | 0.8502 | 0.0977 | **0.8611** | 0.0264 |
| yeast2vs4 | 0.8609 | 0.0705 | 0.8807 | 0.0655 | 0.8814 | 0.0495 | 0.8787 | 0.0265 | **0.8920** | 0.0661 | 0.8703 | 0.0881 | 0.8762 | 0.0263 |
| yeast05679vs4 | 0.7558 | 0.0445 | 0.7753 | 0.0599 | 0.7691 | 0.1226 | 0.7541 | 0.0644 | **0.8085** | 0.0776 | 0.7408 | 0.0521 | 0.7748 | 0.0747 |
| vowel0 | 0.9489 | 0.0659 | **1.0000** | 0.0000 | 0.9705 | 0.0368 | 0.8878 | 0.0834 | 0.8883 | 0.0743 | 0.9578 | 0.0404 | 0.9228 | 0.0728 |
| glass016vs2 | 0.6593 | 0.1588 | **0.6814** | 0.1793 | 0.5924 | 0.1476 | 0.6348 | 0.1527 | 0.6343 | 0.1596 | 0.6371 | 0.0851 | 0.5479 | 0.0493 |
| glass2 | 0.6242 | 0.1205 | **0.6447** | 0.0987 | 0.5687 | 0.1716 | 0.5206 | 0.0763 | 0.6190 | 0.1472 | 0.6217 | 0.1432 | 0.5878 | 0.1392 |
| shuttle0vs4 | 0.9960 | 0.0089 | 0.9960 | 0.0089 | 0.9960 | 0.0089 | 0.9920 | 0.0110 | **0.9997** | 0.0007 | **0.9997** | 0.0007 | 0.9997 | 0.0007 |
| yeast1vs7 | 0.6895 | 0.0928 | 0.5000 | 0.0000 | 0.6754 | 0.1067 | 0.7196 | 0.1150 | 0.6773 | 0.0604 | 0.6583 | 0.0903 | **0.7576** | 0.0791 |
| ecoli4 | **0.9374** | 0.0684 | 0.9171 | 0.0689 | 0.8921 | 0.0956 | 0.8202 | 0.1138 | 0.8120 | 0.1317 | 0.8842 | 0.1046 | 0.8639 | 0.0889 |
| abalone9vs18 | 0.7235 | 0.0930 | 0.6820 | 0.0814 | 0.6896 | 0.1093 | 0.6177 | 0.0971 | 0.6657 | 0.1130 | 0.7063 | 0.0881 | 0.7006 | 0.0757 |
| glass016vs5 | 0.7300 | 0.2433 | 0.8771 | 0.2191 | 0.8443 | 0.2229 | 0.9300 | 0.1008 | 0.8943 | 0.0412 | 0.9486 | 0.0217 | **0.9686** | 0.0235 |
| shuttle2vs4 | 0.9500 | 0.1118 | **1.0000** | 0.0000 | 0.9877 | 0.0185 | 0.9877 | 0.0185 | 0.9875 | 0.0186 | 0.9958 | 0.0093 | 0.9917 | 0.0186 |
| yeast1458vs7 | 0.6169 | 0.0903 | **0.6390** | 0.0778 | 0.5524 | 0.0806 | 0.6344 | 0.0959 | 0.5798 | 0.1053 | 0.6315 | 0.0962 | 0.5351 | 0.0873 |
| yeast4 | 0.8018 | 0.0604 | 0.7242 | 0.0593 | 0.7820 | 0.0665 | 0.5941 | 0.0576 | **0.8217** | 0.0608 | 0.7642 | 0.0905 | 0.7486 | 0.0877 |
| ecoli0137vs26 | 0.8445 | 0.2216 | 0.8281 | 0.2087 | 0.7336 | 0.2414 | 0.7818 | 0.2700 | 0.8062 | 0.2028 | 0.8208 | 0.2099 | 0.8172 | 0.2013 |
| yeast6 | 0.8566 | 0.1034 | 0.7998 | 0.1200 | 0.8315 | 0.1049 | 0.6704 | 0.0630 | **0.8539** | 0.0970 | 0.7922 | 0.0887 | 0.8008 | 0.1052 |
| abalone19 | 0.5268 | 0.1232 | 0.5176 | 0.0385 | 0.5166 | 0.0585 | 0.4994 | 0.0755 | **0.6466** | 0.1201 | 0.5303 | 0.0454 | 0.5401 | 0.0449 |
| Average | **0.8268** | 0.0679 | 0.8218 | 0.0556 | 0.8169 | 0.0691 | 0.7762 | 0.0675 | 0.8202 | 0.0635 | 0.8258 | 0.0613 | 0.8249 | 0.0570 |

**Table 6**
Average Friedman Rankings of and APVs using Holm's procedure in AUC: Data sets preprocessed with SMOTE.

| Algorithm | AUC Ranking | AUC APV |
|---|---|---|
| EGIS-CHC | 3.500 | – |
| SMOTE + PART | 3.555 | 1.00000 |
| SMOTE + 1NN | 3.611 | 1.00000 |
| SMOTE + RIPPER | 3.722 | 1.00000 |
| SMOTE + INNER | 3.889 | 1.00000 |
| SMOTE + BNGE | 4.014 | 1.00000 |
| SMOTE + RISE | 5.708 | 0.00009 |

### *6.3. Analysis of the complexity of the models obtained: number of generalized examples and rules*

Table 7 collects the average number of generalized examples, in the case of EGIS-CHC, BNGE, RISE and INNER; and the average number of rules, in the case of RIPPER and PART. Except for EGIS-CHC, two columns are showed for each method. The column denoted by *none* corresponds to the case in which no preprocessing technique is used previous to the algorithm and the column denoted by *SMOTE* corresponds to the combination of SMOTE with the algorithm in question.

Observing Table 7, we can stress that the lowest number of generalized examples is achieved by EGIS-CHC, although the PART algorithm obtains few rules. Usually, the use of SMOTE leads to increase the number of generalized examples/rules, except in the case of INNER.

### *6.4. Global analysis of results*

Finally, we can make a global analysis of results combining the results offered by Tables from 3–7 and Fig. 3:

- Our proposal, EGIS-CHC, is the best performing one when the data sets are no preprocessed. It outperforms the rest of methods and this hypothesis is confirmed by nonparametric statistical tests.
- The combination of SMOTE with EGIS-CHC produces negative effects by reducing the accuracy in test. EGIS-CHC is a robust algorithm capable to find accurate generalized examples from the original data and it does not require to use preprocess data.
- When the data is treated with SMOTE in the rest of algorithms, an improvement in accuracy is expected. However, they are not able to outperform EGIS-CHC even when SMOTE is applied before. In this sense, the hypotheses of equality cannot be statistically rejected indicating that EGIS-CHC obtains competitive results.
- Considering that EGIS-CHC behaves similarly than other methods combined with SMOTE (1NN, BNGE, INNER, PART and RIPPER), we can emphasize that it requires a lower number of generalized examples or rules than them. Thus, the complexity of the models obtained is lesser assuming that we are measuring it in terms of number generalization/rules. We known that the complexity topic can be very subjective and we do not want to go into detail of types of rules and representations.
- Finally, we can see curious behaviors in some NGE learning methods when they are combined with SMOTE. Considering

**Table 7**
Number of generalized examples obtained by NGE approaches and number of rules produced by RIPPER and PART with and without SMOTE.

| | EGIS-CHC | SMOTE + BNGE | | SMOTE + RISE | | SMOTE + INNER | | SMOTE + RIPPER | | SMOTE + PART | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | None | None | SMOTE | None | SMOTE | None | SMOTE | None | SMOTE | None | SMOTE |
| glass1 | 8.6 | 64.4 | 95.2 | 56.2 | 61.4 | 22.4 | 16.4 | 11.6 | 12.2 | **6.0** | 7.6 |
| ecoli0vs1 | **2.8** | 8.6 | 9.4 | 49.8 | 55.2 | 4.8 | 5.4 | 3.6 | 3.2 | 3.6 | 4.2 |
| wisconsin | **3.0** | 52.4 | 83.0 | 135.0 | 193.8 | 9.2 | 7.2 | 9.2 | 9.2 | 9.4 | 9.6 |
| pima | 14.6 | 282.4 | 413.6 | 360.4 | 415.6 | 10.0 | 8.6 | 24.0 | 25.8 | **7.8** | 8.6 |
| iris0 | **2.0** | **2.0** | **2.0** | 8.6 | 4.0 | 3.2 | 3.8 | 2.4 | 2.2 | **2.0** | **2.0** |
| glass0 | 9.2 | 62.6 | 81.0 | 60.6 | 75.6 | 36.4 | 10.4 | 9.4 | 9.4 | **6.4** | 7.4 |
| yeast1 | 14.8 | 711.2 | 1138.6 | 698.2 | 786.8 | 15.2 | 12.6 | 33.4 | 26.8 | **10.4** | 14.2 |
| haberman | 7.4 | 177.8 | 295.6 | 116.4 | 157.6 | 14.6 | 10.6 | 18.2 | 16.4 | **3.4** | 4.4 |
| vehicle2 | 15.4 | 141.2 | 187.4 | 123.0 | 156.2 | 27.6 | 24.8 | 9.8 | 9.4 | **9.4** | 12.0 |
| vehicle3 | 16.2 | 278.8 | 568.6 | 252.8 | 303.6 | 24.6 | 16.4 | 28.8 | 31.4 | **11.2** | 22.6 |
| glass0123vs456 | **4.0** | 18.6 | 31.2 | 19.8 | 23.6 | 6.4 | 9.0 | 5.8 | 5.2 | 5.0 | 6.6 |
| vehicle0 | **12.2** | 203.2 | 321.6 | 135.4 | 163.2 | 36.4 | 12.4 | 12.2 | 14.2 | 14.0 | 17.2 |
| ecoli1 | **4.6** | 61.8 | 93.2 | 92.2 | 115.0 | 10.2 | 11.6 | 9.8 | 10.8 | 5.0 | 7.2 |
| new-thyroid1 | **2.4** | 11.2 | 14.2 | 21.2 | 18.6 | 4.4 | 7.4 | 3.6 | 4.6 | 4.0 | 4.6 |
| new-thyroid2 | **2.2** | 11.0 | 18.2 | 27.0 | 30.4 | 3.6 | 7.4 | 4.4 | 3.8 | 4.2 | 4.6 |
| ecoli2 | **5.0** | 58.4 | 89.0 | 84.0 | 98.6 | 14.8 | 8.4 | 8.4 | 10.2 | 5.8 | 8.8 |
| segment0 | 9.6 | 139.6 | 144.6 | 100.4 | 94.6 | 46.2 | 13.0 | **7.0** | **7.0** | 7.2 | 7.8 |
| glass6 | **3.0** | 16.2 | 34.0 | 25.0 | 33.2 | 11.0 | 11.2 | 4.0 | 5.2 | 3.2 | 5.6 |
| yeast3 | 11.4 | 245.2 | 538.8 | 597.8 | 652.2 | 30.0 | 9.6 | 20.6 | 26.2 | **9.0** | 10.6 |
| ecoli3 | **4.4** | 43.8 | 93.8 | 62.4 | 72.0 | 17.0 | 9.2 | 7.6 | 8.4 | **4.4** | 6.8 |
| yeast2vs4 | **5.2** | 39.2 | 164.0 | 155.6 | 214.4 | 29.0 | 6.2 | 7.0 | 16.0 | 5.4 | 11.8 |
| yeast05679vs4 | 9.0 | 95.4 | 309.4 | 192.8 | 259.4 | 26.6 | **4.2** | 17.0 | 22.0 | 8.0 | 12.4 |
| vowel0 | 14.2 | 58.4 | 118.2 | 25.6 | 62.6 | 42.2 | 9.4 | 6.6 | 7.2 | **5.4** | 6.8 |
| glass016vs2 | 6.2 | 39.0 | 120.6 | 53.0 | 64.4 | 6.0 | 16.0 | 6.4 | 11.0 | **5.8** | 10.4 |
| glass2 | **5.6** | 42.0 | 99.0 | 60.6 | 70.6 | 6.2 | 20.8 | 7.2 | 10.0 | **5.6** | 10.0 |
| shuttle0vs4 | **2.0** | **2.0** | 3.6 | **2.0** | 67.6 | 18.2 | 7.6 | **2.0** | 2.8 | **2.0** | 2.8 |
| yeast1vs7 | **5.8** | 80.8 | 686.4 | 182.6 | 686.4 | 11.2 | 10.4 | 15.0 | 23.2 | 6.8 | 15.8 |
| ecoli4 | **3.0** | 19.6 | 38.2 | 35.0 | 49.6 | 9.8 | 10.6 | 4.4 | 5.4 | 3.8 | 6.4 |
| abalone9vs18 | 18.8 | 115.0 | 425.0 | 125.4 | 180.4 | 13.8 | 14.2 | 15.8 | 25.2 | **7.8** | 26.8 |
| glass016vs5 | 3.8 | 11.0 | 37.4 | 19.8 | 26.8 | 4.6 | 10.2 | **3.2** | 6.0 | 4.0 | 5.4 |
| shuttle2vs4 | 2.4 | **2.0** | 9.4 | 4.0 | 11.4 | 4.4 | 9.4 | 2.8 | 4.4 | 3.0 | 4.2 |
| yeast1458vs7 | 7.4 | 132.2 | 598.2 | 276.0 | 326.6 | 9.2 | 19.6 | 17.0 | 27.6 | **3.6** | 23.0 |
| yeast4 | 6.6 | 178.8 | 794.6 | 531.6 | 597.8 | 26.0 | 7.6 | 19.6 | 25.8 | **6.4** | 20.4 |
| ecoli0137vs26 | **2.0** | 15.8 | 40.0 | 29.4 | 61.4 | 2.0 | 7.2 | 4.2 | 7.4 | 3.0 | 5.8 |
| yeast6 | 7.2 | 145.6 | 415.4 | 440.6 | 503.6 | 14.0 | 7.6 | 12.4 | 13.2 | **4.8** | 13.2 |
| abalone19 | 13.0 | 338.8 | 2207.8 | 764.0 | 1035.4 | 12.6 | 10.0 | 12.6 | 30.2 | **1.0** | 36.0 |
| Average | 7.4 | 108.5 | 286.7 | 164.6 | 214.7 | 16.2 | 10.7 | 10.8 | 13.3 | **5.8** | 10.7 |

RISE, the use of SMOTE does not offer any advantage in terms of accuracy, but it makes difficult its operation and it has to get more generalized examples. Otherwise, INNER is the worst method in imbalanced problems when they are not preprocessed, but the combination SMOTE + INNER offers excellent results and even the number of generalized examples resulted is lower in this last case.

## 7. Concluding remarks

The purpose of this paper is to present EGIS-CHC, an evolutionary model to improve imbalanced classification based on the nested generalized example learning. The proposal performs an optimized selection of previously defined generalized examples obtained by a simple and fast heuristic.

The results show that the use of generalized exemplar selection based on evolutionary algorithms can obtain promising results to optimize the performance in imbalanced domains. It was compared with classical (RISE and BNGE) and recent (INNER) nested generalized learning approaches and two state-of-the-art rule induction methods, RIPPER and PART. EGIS-CHC clearly outperforms all of them when data is not preprocessed. The paper also shows the analysis of using SMOTE as data imbalanced preprocessing and our approach offers similar results in accuracy to the ones offered by the combination of SMOTE with the learning approaches mentioned above, but it requires to retain a lower number of generalized examples, thus yielding simpler models.

## Acknowledgement

This work was supported by TIN2008-06681-C06-01 and TIN2008-06681-C06-02. J. Derrac holds a research scholarship from the University of Granada.

## References

[1] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Machine Learning 6 (1) (1991) 37–66.

[2] F. Al-Obeidat, N. Belacel, J.A. Carretero, P. Mahanti, Differential evolution for learning the classification method proaftn, Knowledge-Based Systems 23 (5) (2010) 418–426.

[3] 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 Multiple-Valued Logic and Soft Computing 17 (2–3) (2011) 255–287.

[4] J. Alcala-Fdez, L. Sánchez, S. García, M. del Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernández, F. Herrera, Keel: a software tool to assess evolutionary algorithms for data mining problems, Soft Computing 13 (3) (2009) 307–318.

[5] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, Sigkdd Explorations 6 (1) (2004) 20–29.

[6] A.P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern Recognition 30 (7) (1997) 1145–1159.

[7] J.R. Cano, F. Herrera, M. Lozano, Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study, IEEE Transactions on Evolutionary Computation 7 (6) (2003) 561–575.

[8] Y.-W. Chang Chien, Y.-L. Chen, Mining associative classification rules with stock trading data – a ga-based method, Knowledge-Based Systems 23 (6) (2010) 605–614.

[9] N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, Smote: synthetic minority oversampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

[10] N. Chawla, D. Cieslak, L. Hall, A. Joshi, Automatically countering imbalance and its empirical relationship to cost, Data Mining and Knowledge Discovery 17 (2) (2008) 225–252.

[11] N.V. Chawla, N. Japkowicz, A. Kotcz, Editorial: special issue on learning from imbalanced data sets, SIGKDD Explorations Newsletter 6 (1) (2004) 1–6.

[12] W.W. Cohen, Fast effective rule induction, in: Proceedings of the Twelfth International Conference on Machine Learning, 1995, pp. 115–123.

[13] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27.

[14] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[15] J. Derrac, S. García, F. Herrera, IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule, Pattern Recognition 43 (6) (2010) 2082–2105.

[16] P. Domingos, Unifying instance-based and rule-based induction, Machine Learning 24 (1996) 141–168.

[17] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer-Verlag, 2003.

[18] L.J. Eshelman, The CHC adaptive search algorithm: how to safe search when engaging in nontraditional genetic recombination, in: G.J.E. Rawlings (Ed.), Foundations of genetic algorithms, 1991, pp. 265–283.

[19] A. Fernández, M.J. del Jesus, F. Herrera, On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced datasets, Information Sciences 180 (8) (2010) 1268–1291.

[20] A. Fernández, S. García, J. Luengo, E. Bernadó-Mansilla, F. Herrera, Genetics-based machine learning for rule induction: Taxonomy, experimental study and state of the art, in: IEEE Transactions on Evolutionary Computation 4 (6) (2010) 913–941.

[21] E. Frank, I.H. Witten, Generating accurate rule sets without global optimization, in: Proceedings of the Fifteenth International Conference on Machine Learning, ICML, 1998, pp. 144–151.

[22] A.A. Freitas, Data Mining and Knowledge Discovery with Evolutionary Algorithms, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[23] J. Fürnkranz, Separate-and-conquer rule learning, Artificial Intelligence Review 13 (1) (1999) 3–54.

[24] S. García, J.R. Cano, F. Herrera, A memetic algorithm for evolutionary prototype selection: a scaling up approach, Pattern Recognition 41 (8) (2008) 2693–2709.

[25] S. García, A. Fernández, F. Herrera, Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems, Applied Soft Computing 9 (4) (2009) 1304–1314.

[26] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, Soft Computing 13 (10) (2009) 959–977.

[27] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Information Sciences 180 (2010) 2044–2064.

[28] S. García, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.

[29] S. García, F. Herrera, Evolutionary under-sampling for classification with imbalanced data sets: proposals and taxonomy, Evolutionary Computation 17 (3) (2009) 275–306.

[30] J. Grzymala-Busse, J. Stefanowski, S. Wilk, A comparison of two approaches to data mining from imbalanced data, Journal of Intelligent Manufacturing 16 (6) (2005) 565–573.

[31] C. Guerra-Salcedo, S. Chen, D. Whitley, S. Smith, Fast and accurate feature selection using hybrid genetic strategies, in: CEC, 1999, pp. 177–184.

[32] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.

[33] E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas, A.C.P.L.F. De Carvalho, A survey of evolutionary algorithms for clustering, IEEE Transactions on Systems Man and Cybernetics. Part C 39 (2) (2009) 133–155.

[34] I. Kononenko, M. Kukar, Machine Learning and Data Mining: Introduction to Principles and Algorithms, Horwood Publishing Limited, 2007.

[35] M. Laguia, J.L. Castro, Local distance-based classification, Knowledge-Based Systems 21 (7) (2008) 692–703.

[36] S. Li, H. Wu, D. Wan, J. Zhu, An effective feature selection method for hyperspectral image classification based on genetic algorithm and support vector machine, Knowledge-Based Systems 24 (1) (2011) 40–48.

[37] H. Liu, H. Motoda (Eds.), Instance Selection and Construction for Data Mining, Springer, 2001.

[38] O. Luaces, A. Bahamonde, Inflating examples to obtain rules, International Journal of Intelligent Systems 18 (11) (2003) 1113–1143.

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

[40] D. Medin, M. Schaffer, Context theory of classification learning, Psychological Review 85 (1978) 207–238.

[41] A. Orriols-Puig, E. Bernadó-Mansilla, Evolutionary rule-based systems for imbalanced data sets, Soft Computing 13 (3) (2009) 213–225.

[42] D. Pyle, Data Preparation for Data Mining, The Morgan Kaufmann Series in DMS, 1999.

[43] S. Salzberg, A nearest hyperrectangle method, Machine Learning 6 (1991) 151–276.

[44] S. Senthamarai Kannan, N. Ramaraj, A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm, Knowledge-Based Systems 23 (6) (2010) 580–585.

[45] D. Sheskin, Handbook of parametric and nonparametric statistical procedures, Chapman & Hall/CRC, 2006.

[46] I. Triguero, S. García, F. Herrera, Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification, Pattern Recognition 44 (4) (2011) 901–916.

[47] I. Triguero, S. García, F. Herrera, IPADE: Iterative prototype adjustment for nearest neighbor classification, IEEE Transactions on Neural Networks 21 (12) (2010) 1984–1990.

[48] D. Wettschereck, T.G. Dietterich, An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms, Machine Learning 19 (1995) 5–27.

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

[50] S. Zhang, Cost-sensitive classification with respect to waiting cost, Knowledge-Based Systems 23 (5) (2010) 369–378.