Applied Soft Computing 11 (2011) 3032–3045

Contents lists available at ScienceDirect

Applied Soft Computing

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

Evolutionary selection of hyperrectangles in nested generalized exemplar learning

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

[a] Department of Computer Science, University of Jaén, 23071, Jaén, Spain
[b] Department of Computer Science and Artificial Intelligence, University of Granada, 18071, Granada, Spain

ARTICLE INFO

Article history:
Received 7 February 2010
Received in revised form 7 July 2010
Accepted 29 November 2010
Available online 16 December 2010

Keywords:
Nested generalized exemplar
Nearest neighbor
Hyperrectangles
Evolutionary algorithms
Data reduction
Classification

ABSTRACT

The nested generalized exemplar theory accomplishes learning by storing objects in Euclidean *n*-space, as hyperrectangles. Classification of new data is performed by computing their distance to the nearest "generalized exemplar" or hyperrectangle. This learning method allows the combination of the distance-based classification with the axis-parallel rectangle representation employed in most of the rule-learning systems. In this paper, we propose the use of evolutionary algorithms to select the most influential hyperrectangles to obtain accurate and simple models in classification tasks. The proposal has been compared with the most representative models based on hyperrectangle learning; such as the BNGE, RISE, INNER, and SIA genetics based learning approach. Our approach is also very competitive with respect to classical rule induction algorithms such as C4.5Rules and RIPPER. The results have been contrasted through non-parametric statistical tests over multiple data sets and they indicate that our approach outperforms them in terms of accuracy requiring a lower number of hyperrectangles to be stored, thus obtaining simpler models than previous NGE approaches. Larger data sets have also been tackled with promising outcomes.

1. Introduction

Exemplar-based learning was originally proposed in [1] and considers a set of methods widely used in machine learning and data mining [2,3]. A similar scheme for learning from examples is based on the Nested generalized exemplar (NGE) theory. It was introduced in [4] 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) [5] and were proposed in order to extend it. NGE learning algorithms are very popular for their simplicity and efficient results.

In NGE theory, generalizations take the form of hyperrectangles in an Euclidean *n*-space. It can be approached as an exemplar-based generalization model. The hyperrectangles may be nested and inner hyperrectangles serve as exceptions to surrounding hyperrectangles. An specific example can be viewed as a minimal hyperrectangle. Hyperrectangles are axis-parallel rectangle representations employed in most of the rule-learning systems [6]. After the learning process, a new example can be classified by computing the Euclidean distance between the example and each of the hyperrectangles, predicting the class of the new example considering the nearest hyperrectangle. If two or more hyperrectangles cover the example, a conflict resolution method to determine the predicted class has to be used [4].

Several works argue the benefits of using hyperrectangles together with instances to form the classification rule [7–9]. With respect to instance-based classification [1], the employment of hyperrectangles 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 [6], 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 us to capture generalizations with exceptions.

The methods used for generating nearest hyperrectangles classification can work in an incremental fashion, such as EACH [4], or in batch mode (BNGE [7], RISE [8], FAN [10] and INNER [9]). The incremental way is dependent on the order of presentation of examples and usually offers poor results in standard classification. However, it could be used in on-line learning scenarios. Batch mode meth-

* Corresponding author. Tel.: +34 953 212802.
E-mail addresses: sglopez@ujaen.es (S. García), jderrac@decsai.ugr.es (J. Derrac), julianlm@decsai.ugr.es (J. Luengo), ccarmona@ujaen.es (C.J. Carmona), herrera@decsai.ugr.es (F. Herrera).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.11.030

ods employ heuristics to determine the choice of the exemplars to be merged or generalized at each stage. The results offered are very interesting and they usually outperform the results obtained by the NN classifier [7].

Extensions to NGE can be found in the specialized literature. Heath et al. [11] address the problem of whether reducing the memory capacity of a learning algorithm will have an effect on the speed learning, for a particular concept class, that of nested hyperrectangles. In [12], authors investigate the impact on the predictive accuracy of the learnt concepts by NGE as a consequence of using three distance functions, namely HVDM, IVDM and WVDM [13]. The fuzzy NGE model was proposed in [14,15] and the transformation of neural networks based knowledge to NGE based knowledge was investigated in [16,17]. An interesting study for analyzing hybridizations of exemplar-based learning with other machine learning paradigms can be found in [18].

The problem of yielding an optimal number of hyperrectangles for classifying a set of points is NP-hard. A large but finite subset of hyperrectangles can be easily obtained following a simple heuristic algorithm acting over the training data. However, almost all hyperrectangles produced could be irrelevant and, as a result, most influential ones must be distinguished. This complete set of hyperrectangles is thus suitable for improvement by a data reduction technique [19]. Evolutionary algorithms (EAs) [20] have been used for data reduction with promising results [21,22]. They have been successfully used for feature selection [23–27], instance selection [28–31], simultaneous instance and feature selection [32,33] and under-sampling for imbalanced learning [34,35]. NGE is also directly related to clustering and EAs have been extensively used for this problem [36]. EAs for clustering could be useful as alternative components of NGE learning, especially when the initial candidate set of hyperrectangles to be selected has to be obtained.

In this paper, we propose the use of EAs for hyperrectangles' selection in classification tasks. One similar approach is SIA [37], which is a genetics-based machine learning method to obtain a set of rules by means of computing distances among rules. Our objective is to increase the accuracy of this type of representation by means of selecting the best suitable set of hyperrectangles which optimizes the nearest hyperrectangle classification rule. We compare our approach with other NGE learning models, such as BNGE, RISE, INNER and SIA, and two well-known rule induction learning methods: RIPPER and C4.5Rules. The empirical study has been contrasted via non-parametrical statistical testing [38–41]. The results show an improvement in accuracy whereas the number of hyperrectangles stored in the final subset is reduced. This outcome is especially observed when dealing with larger data sets. Regarding classic rule induction, we can observe that our proposal is more adaptable under different types of input data.

We note that the proposal described in this paper is an extended algorithm to that described in our previous work [42]. Our previous version presented some weaknesses related to the few number of examples covered by the hyperrectangles learned and the treatment of noisy examples. In this paper, the coverage of the hyperrectangles is incorporated in the proposal and we present a modification based on a previous stage of noise filtering. In addition, more data sets (including large size data sets) and appropriate statistical tools have been used to justify the conclusions achieved.

The paper is organized as follows. Section 2 gives an explanation of the NGE learning model. In Section 3, all topics concerning the approach proposed to tackle this problem are explained. In Section 4 the experimentation framework is given and in Section 5 the results and analysis are presented. In Section 6, the conclusions are highlighted. Finally, Appendix A is included in order to illustrate the comparisons of our proposal with other techniques through star plots.

2. NGE learning model

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 [4]. 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 subsequent 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, 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 two advanced approaches: INNER in Section 2.3.1 and genetics-based SIA in Section 2.3.2.

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 hyperrectangle). For the remainder of this paper, we will refer to the example to be classified as E and the hyperrectangle as H, independently of whether H is formed by a single point or it has some volume.

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

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

where

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

M is the number of attributes of the data, E_{f_i} is the value of the ith feature of the example, H_{upper} and H_{lower} are the upper and lower values of H for a specific attribute and $\max_i$ and $\min_i$ are the maximum and minimum values for ith feature in training data, respectively.

The distance measured by this formula is equivalent to the length of a line dropped perpendicularly from the point E_{fi} to the nearest surface, edge or corner of H. Note that points internal to a hyperrectangle have distance 0 to that rectangle. In the case of overlapping rectangles, several strategies could be followed, but the most usual is that in which a point falling in the area of overlap belongs to the smaller rectangle. The size of a hyperrectangle is defined in terms of volume. In nominal attributes, the distance is 0 when two attributes have the same categorical label, and l on the contrary.

NGE theory also refers to weights associated with features in examples, but they are not considered in this paper because they can be used independently to the induction of hyperrectangles in NGE. In [7], the authors noted that the use of weights do not always improve the performance of a NGE learner. Actually, they showed that mutual information weights could be appropriate in most of the cases. Nevertheless, they insisted on that the mechanism of feature weighting is independent of NGE and can therefore be used as a pre-processing step for any inductive learning algorithm.

2.2. *Classical proposals*

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.

2.2.1. *BNGE: batch nested generalized exemplar*

BNGE is a batch version of the first NGE model (also known as EACH [4]) and it is proposed to alleviate some drawbacks presented in the initial NGE [7]. It changes its incremental fashion to a batch mode and adds some modifications in the matching rule, such as including all possible nominal values in hyperrectangle definition and adding a mechanism to deal with missing values. The generalization of a hyperrectangle is performed by expanding its frontiers to just cover the desired example.

BNGE only merges hyperrectangles if the new generalized hyperrectangle does not cover (or does not overlap with) any hyperrectangles from any other classes. It does not permit overlapping or nesting, which are two of the identified disadvantages of incremental NGE.

2.2.2. *RISE: unifying instance-based and rule-based induction*

RISE [8] 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 to those explained above, but it also introduces some improvements regarding distance computations, since the SVDM distance [13] is used in nominal attributes. RISE selects the rule with the highest accuracy (using the Laplace correction used by many existing rule-induction techniques [6]) instead of choosing the smallest rule that covers the example.

BNGE and RISE follow a similar mechanism to produce hyperrectangles. They start from the complete training set and try to merge the nearest examples/hyperrectangles if the global accuracy is not decreased. RISE uses a leave-one-out methodology to compute training accuracy and, contrary to BNGE, nesting or overlapping between hyperrectangles is allowed.

2.3. *Advanced proposals*

As advanced proposals, we describe INNER and SIA NGE learning models.

2.3.1. *INNER: inflating examples to obtain rules*

INNER [9] is a machine learning system which induces hyperrectangles from a set of training examples, aiming to obtain a final rule set which represents properly the domain of the problem.

It 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. These movements are accepted only if they increase the confidence of a successful classification when comparing it against the unconditional classification by using the class attribute of the hyperrectangle. Then, it applies a set of elastic transformations to the hyperrectangles, to finally obtain a concise and accurate hyperrectangle set to classify new data. These transformations include several processes:

- selection of the most appropriate hyperrectangle.
- the pruning of some of its conditions.
- two extension processes, where intersections between hyperrectangles are allowed only in the second one.

Finally, if the set of hyperrectangles still does not achieve enough coverage level over the training examples, a final hyperrectangle generation process is carried out again. Each hyperrectangle obtained can be seen as a representative cluster of training examples of the same class. Thus, when a hyperrectangle is used to classify an unseen example, the decision is supported by a natural and solid explanation.

2.3.2. *SIA: genetics-based supervised induction algorithm*

SIA [37] is a classical iterative rule learning method [43] based on genetic algorithms. The algorithm induces a set of rules which can be considered as hyperrectangles in NGE. We define below its main characteristics:

- The rule set is represented by classical IF-THEN rules. The conditions of the different attributes of a rule may have a “don’t care” value, a pair attribute-value if the attribute is symbolic, or an interval value $[B,B']$ if the attribute is numerical.
- The main procedure of SIA is detailed as follows: first it selects an uncovered sample and then it builds the most specific rule that matches that example. Then, it generalizes the condition part of the rule using a GA and it labels “covered” all the examples matched by this rule and adds it to the rule set. This process is repeated until no more examples remain uncovered.
- To classify a new pattern E we compute the distance of each rule H to the example as D_{EH}, as we have explained above. If more than one rule has the same minimal distance to E, then the rule used to predict the class is the one with highest confidence value.

3. Evolutionary selection of hyperrectangles

The approach proposed in this paper, named evolutionary hyperrectangle selection by CHC (*EHS-CHC*), is fully explained in this section. First, we introduce the CHC model used as an EA to perform hyperrectangle selection in Section 3.1. After this, the specific issues regarding representation and fitness function are specified in Section 3.2. Section 3.3 describes the process for generating the initial set of hyperrectangles and Section 3.4 presents the extended version of *EHS-CHC*: *filtered EHS-CHC*.

3.1. *CHC model*

As an evolutionary computation method, we have used the CHC model [44,29]. 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.

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 a form of heterogeneous recombination using HUX, a special recombination operator [44]. HUX exchanges half of the bits that differ between parents, where the bit position to be exchanged is randomly determined. CHC also employs a method of incest prevention. Before applying HUX to the 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. The initial threshold is set at $L/4$, where L is the length of the chromosomes. If no offspring are inserted into the new population then the threshold is reduced by one.

No mutation is applied during the recombination phase. Instead, when the population converges or the search stops making progress (i.e., the difference threshold has dropped to zero and no new offspring are being generated which are better than any member of the parent population) the population is reinitialized to introduce new diversity to the search. The chromosome representing the best solution found over the course of the search is used as a template to reseed the population. Reseeding of the population is accomplished by randomly changing 35% of the bits in the template chromosome to form each of the other $N-1$ new chromosomes in the population. The search is then resumed.

The pseudocode of CHC appears in Algorithm 1.

Algorithm 1 *(Pseudocode of CHC algorithm).*

```
input  : A population of chromosomes $P_a$
output : An optimized population of chromosomes $P_a$

t ← 0;
Initialize($P_a$,ConvergenceCount);
while not EndingCondition(t,$P_a$) do
    Parents ← SelectionParents($P_a$);
    Offspring ← HUX(Parents);
    Evaluate(Offspring);
    $P_n$ ← ElitistSelection(Offspring,$P_a$);
    if not modified($P_a$,$P_n$) then
        ConvergenceCount ← ConvergenceCount − 1;
        if ConvergenceCount = 0 then
            $P_n$ ← Restart($P_a$);
            Initialize(ConvergenceCount);
        end
    end
    t ← t + 1;
    $P_a$ ← $P_n$;
end
```

3.2. Representation and fitness function

Let us assume that there is a training set *TR* with *P* instances which consists of pairs (x_i, y_i), $i = 1, ..., P$, where x_i defines an input vector of attributes and y_i defines the corresponding class label. Each one of the *P* instances has *M* input attributes. Let us also assume that there is a hyperrectangle set *HS* with *N* hyperrectangles whose rule representation consists of pairs (H_i, y_i), $i = 1, ..., N$, where H_i defines a set of conditions $(A_1, A_2, ..., A_M)$ and y_i defines the corresponding class label. Each one of the *N* hyperrectangles has *M* conditions which can be numerical conditions, expressed in terms of minimum and maximum values in intervals [0,1]; or they can be categorical conditions, by using a set of possible values $A_i = \{v_{1i}, v_{2i}, \ldots, v_{vi}\}$, assuming that it has *vi* different values. Note that we make no distinction between a hyperrectangle with volume and minimal hyperrectangles formed by isolated points. Let $S \subseteq HS$ be the subset of selected hyperrectangles that result from the run of a hyperrectangle selection algorithm.

Hyperrectangle selection can be considered as a search problem to 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 *HS*. This is accomplished by using a binary representation. A chromosome consists of *N* genes (one for each hyperrectangle in *HS*) with two possible states: 0 and 1. If the gene is 1, its associated hyperrectangle is included in the subset of *HS* represented by the chromosome. If it is 0, this does not occur.
- *Fitness function:* Let *S* be a subset of hyperrectangles of *HS* and be coded by a chromosome. We define a fitness function based on the accuracy (classification rate) evaluated over *TR* through the rule described in Section 2.1.

$$Fitness(S) = \beta \cdot (\alpha \cdot clas\_rat + (1-\alpha) \cdot perc\_red) + (1-\beta) \cdot cover.$$

clas_rat denotes the percentage of correctly classified objects from *TR* using *S*. *perc_red* is defined as

$$perc\_red = 100 \cdot \frac{|HS| - |S|}{|HS|},$$

and *cover* denotes the total coverage of examples in *TR* in the subset of selected hyperrectangles, or in other words, the number of examples of *TR* whose distance computation has been equal to 0 (examples covered by hyperrectangles).

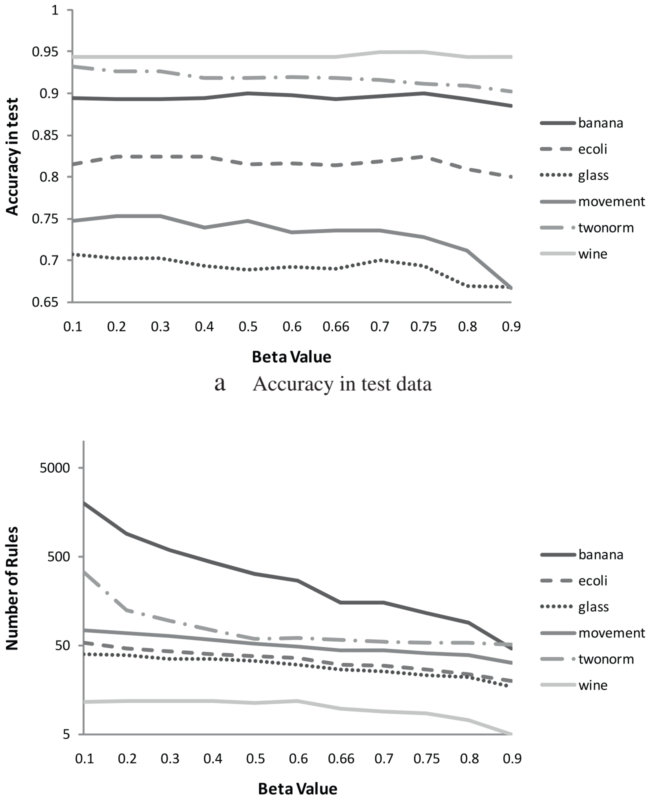
The objective of the EAs is to maximize the fitness function defined, i.e., maximize the classification rate and coverage while minimizing the number of hyperrectangles selected. Although the fitness function defined is focused on discarding single trivial hyperrectangles (points), exceptions could be presented in special cases where points are necessary to achieve high rates of accuracy. Thus, the necessity of using single hyperrectangles or not will be determined by the tradeoff accuracy-coverage and will be conditioned by the problem tackled.

Regarding the parameters, we preserved the value of $\alpha = 0.5$ as the best choice, due to the fact that it was analyzed in previous works related to instance selection [28–30]. We have determined empirically that a suitable value for β should fall between 0.5 and 0.75. This empirical study is presented in the following.

Fig. 1 depicts a graphical representation of the performance of *EHS-CHC* in accuracy in test and number of rules yielded in six data sets of different characteristics (see Table 1). In Fig. 1, the accuracy in test data obtained by *EHS-CHC* is represented considering the values of β from 0.1 to 0.9 and in Fig. 1, the number of rules yielded are represented following the same mechanism. As we can see, the trend in the accuracy lines is to start decreasing when $\beta = 0.75$ in most of cases, except in *wine* where no changes in accuracy can be highlighted due to the fact that this data set is very simple. Furthermore, in a certain subset of data sets, this decrease is noticed before, when $\beta = 0.6$. On the other hand, the lines of number or rules yielded by *EHS-CHC* follow a decremental behavior according to β increases. For these reasons, we decided to choose $\beta = 0.66$ as a general suitable value.

We want to notice that our objective is to identify the best values of the parameters that configure the evolutionary approach in a general way. For a specific classification problem, these values could be tuned in order to optimize the results achieved, but this may affect to other aspects, like efficiency or simplicity. General rules can be given about this topic:

- The number of evaluations and population size are the main factors for obtaining good results in accuracy and simplicity. The increase of these values has a negative effect on efficiency. In larger problems, it may be necessary to increase both values, but we will show in the experimental study that values of 10,000 and 50 work appropriately, respectively.



a Accuracy in test data

b Number of rules

Fig. 1. Analysis of the effect of the parameter β.

- Parameters α and β allow us to obtain a desired trade-off between the accuracy and the number of rules. In the case of obtaining poor accuracy rates in a specific problem we have to increase α or decrease β. In contrary case, when the rules obtained are numerous and we are interested in obtaining simpler models, we have to increase β or decrease α.

The same mechanisms to perform a classification of an unseen example shown in [4] are used in our approach. 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.

Our approach computes the volume of a hyperrectangle in the following way:

$$V_H = \prod_{i}^{M} L_i$$

where L_i is computed for each condition as

$$L_i = \begin{cases} H_{\text{upper}} - H_{\text{lower}} \text{ if numeric and } H_{\text{upper}} \neq H_{\text{lower}} \\ 1 \text{ if numeric and } H_{\text{upper}} = H_{\text{lower}} \\ \dfrac{\text{num. values selected}}{vi} \text{ if nominal} \end{cases}$$

Table 1
Summary description of used data sets.

Data set	#Ex.	#Atts.	#Num.	#Nom.	#Cl.
Appendicitis	106	7	7	0	2
Australian credit	690	14	8	6	2
Balance	625	4	4	0	3
Breast	286	9	0	9	2
Bupa	345	6	6	0	2
Cleveland	297	13	13	0	5
Contraceptive	1473	9	9	0	3
Crx	125	15	6	9	2
Dermatology	366	34	34	0	6
E. coli	336	7	7	0	8
German	1,000	20	7	13	2
Glass	214	9	9	0	7
Haberman	306	3	3	0	2
Hepatitis	155	19	19	0	2
Iris	150	4	4	0	3
Led7digit	500	7	7	0	10
Lymphography	148	18	3	15	4
Mammographic	961	5	0	5	2
Movement	360	90	90	0	15
Newthyroid	215	5	5	0	3
Pima	768	8	8	0	2
Saheart	462	9	8	1	2
Sonar	208	60	60	0	2
Spectfheart	267	44	44	0	2
Tic-tac-toe	958	9	0	9	2
Vehicle	846	18	18	0	4
Wine	178	13	13	0	3
Wisconsin	683	9	9	0	2
Yeast	1484	8	8	0	10
Zoo	101	16	0	16	7
Abalone	4174	8	7	1	28
Banana	5300	2	2	0	2
Coil2000	9822	85	85	0	2
Magic	19,020	10	10	0	2
Page-blocks	5472	10	10	0	5
Penbased	10,992	16	16	0	10
Phoneme	5404	5	5	0	2
Satimage	6435	36	36	0	7
Segment	2310	19	19	0	7
Splice	3190	60	0	60	3
Texture	5500	40	40	0	11
Twonorm	7400	20	20	0	2

3.3. Obtention of the initial set of hyperrectangles

There is a detail not specified yet. It refers to the creation of the initial set of hyperrectangles. In our approach, we have used a simple heuristic which is fast and obtains acceptable results. The heuristic yields a hyperrectangle from each example in the training set. For each one, it finds the $K-1$ nearest neighbors being the Kth neighbor an example of a different class. Then, each hyperrectangle 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 hyperrectangles are obtained, the duplicated ones are removed, keeping one representative in each case. Hence $|HS| \leq |TR|$. Note that point hyperrectangles are possible to be obtained using this heuristic when the nearest neighbor of an instance belongs to a different class.

3.4. An Extended version of EHS-CHC: filtered EHS-CHC

It is known that real data may contain noise or erroneous data. Noisy data could be detrimental when using the heuristic for creating the initial set of hyperrectangles explained above. In some cases, a larger hyperrectangle could be divided into two or more smaller hyperrectangles when a noisy datum is found in the middle

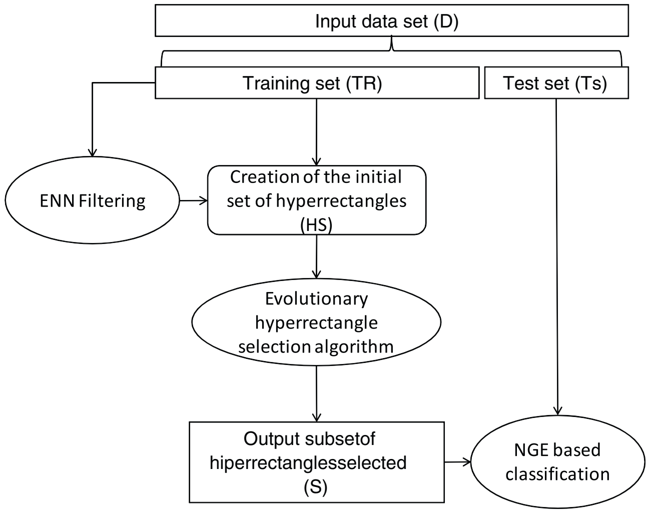

Input data set (D)
Training set (TR)
Test set (Ts)
ENN Filtering
Creation of the initial set of hyperrectangles (HS)
Evolutionary hyperrectangle selection algorithm
Output subsetof hiperrectanglesselected (S)
NGE based classification

Fig. 2. Evolutionary hyperrectangle selection process.

of clean data. In order to avoid this behavior, we include a filtering noise stage based on the ENN algorithm [45] prior to the creation of the initial set of hyperrectangles. ENN starts working with the original training data and then it removes, in batch mode, those instances whose class does not agree with the majority of its k nearest neighbors. k is set to a value of 3. This approach will be named *filtered EHS-CHC*. In *filtered EHS-CHC*, a value of k greater than 3 is not appropriate together with the mechanism of generation of the initial set of hyperrectangles (see Section 3.3).

Fig. 2 represents the evolutionary hyperrectangle selection process followed by *EHS-CHC*.

4. Experimental framework

In this section we first provide details of the real-world problems chosen for the experimentation and the configuration parameters of the methods studied (Sections 4.1 and 4.2). Finally, we present the statistical tests applied to compare the results obtained with the different approaches (Section 4.3).

4.1. Data sets

In the experimental study, we selected 42 data sets from the UCI repository [46] and KEEL-data set[1] [47,48]. Table 1 summarizes the properties of these data sets. It shows, for each data set, the number of examples (#Ex.), the number of attributes (#Atts.), the number of numerical (#Num.) and nominal (#Nom.) attributes, and the number of classes (#Cl.). The data sets are grouped into two categories depending on the size they have (a horizontal line divides them in the table). Small data sets have less than 2000 instances and medium data sets have more than 2000 instances. The data sets considered are partitioned using the *ten fold cross-validation* (*10-fcv*) procedure and non-stochastic algorithms have been run three times.

4.2. Parameters

The configuration parameters are shown in Table 2. They are common to all problems, except for the INNER algorithm, which is quite sensitive to the parameters chosen and has two different configurations depending on the size of the problems tackled.

Table 2
Parameter specification for all the methods employed in the experimentation.

Algorithm	Parameters
BNGE	It has not parameters
RISE	$Q = 1$, $S = 2$
INNER small data sets	Initial examples = 10, MaxCycles = 5 Min. coverage = 0.95, Min. Presentations = 3000 Regularize = 300, Threshold = −100
INNER medium data sets	Initial examples = 10, MaxCycles = 5 Min. coverage = 0.95, Min. Presentations = 10000 Regularize = 1000, Threshold = −100
SIA	Iterations = 200, $\alpha = 150$ $\beta = 0$, Threshold Strength = 0
C4.5-Rules	Prune = True, Confidence level = 0.25 Minimum number of item-sets per leaf = 2
Ripper	Size of growing subset = 66% Repetitions of the optimization stage = 2
EHS-CHC	Popul. Size = 50, Num. Evaluations = 10000 $\alpha = 0.5$, $\beta = 0.66$
Filtered EHS-CHC	Popul. Size = 50, Num. Evaluations = 10,000 $\alpha = 0.5$, $\beta = 0.66$, k for ENN = 3

The values of previous proposed approaches were selected according to the recommendation of the corresponding authors of each algorithm, and the values set for our approach have been either empirically determined or follow recommendations from previous applications of the CHC model to other data reduction problems [28,29]. We also refer to the suggestions given in Section 3.2.

4.3. Statistical tests for performance comparison

In this paper, we use the hypothesis testing techniques to provide statistical support for the analysis of the results [49,40]. Indeed, we use non-parametric tests, due to the fact that the initial conditions that guarantee the reliability of the parametric tests may not be satisfied, causing the statistical analysis to lose credibility with the use of parametric tests [38,39]. Specifically, we use the Wilcoxon signed-rank test as a non-parametric statistical procedure for performing pairwise comparisons between two algorithms. For multiple comparisons, we use three non-parametric tests: Friedman, Friedman Aligned-Ranks and Quade tests [41].

These tests are suggested in the studies presented in [38–41]. More information about these tests and other statistical procedures specifically designed for use in the field of Machine Learning can be found at the SCI^2*S* thematic public website on Statistical Inference in Computational Intelligence and DataMining.[2]

5. Results and analysis

This section shows the results obtained in the experimental study as well as the analysis based on them. The experimental study will be divided into two parts: experiments for small data sets (Section 5.1) and experiments for medium data sets (Section 5.2). In addition, a study of the efficiency of the NGE models considered in this paper is carried out in Section 5.3.

5.1. Results and analysis for small data sets

Table 3 shows the results measured by accuracy in test data for each approach considered in this paper. Table 4 shows the average number of rules/hyperrectangles yielded by the approaches considered. For each data set, the mean accuracy or number of rules and the standard deviation (SD) are computed. The best case in each data set is stressed in bold. The last row in each table shows the average considering all data sets.

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

[2] http://sci2s.ugr.es/sicidm.

Table 3
Accuracy results in test over small data sets.

Data sets	1NN		C4.5Rules		RIPPER		BNGE		RISE		INNER		SIA		EHS-CHC		Filtered EHS-CHC	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Appendicitis	0.7936	0.1151	0.8318	0.1120	0.8382	0.1241	0.8609	0.0869	0.8218	0.1010	0.8600	0.0984	0.8418	0.0833	0.8609	0.1059	**0.8691**	0.1155
Australian	0.8145	0.0429	0.8536	0.0277	0.8275	0.0293	0.8551	0.0368	0.8058	0.0444	0.8551	0.0306	0.6087	0.0572	0.7551	0.0466	**0.8580**	0.0296
Balance	0.7904	0.0646	0.8175	0.0415	0.5278	0.0838	0.8192	0.0452	0.4960	0.0200	0.7423	0.0391	**0.8382**	0.0428	0.8223	0.0434	0.7998	0.0461
Breast	0.6535	0.0607	0.6776	0.0761	0.6401	0.0555	0.6229	0.0798	0.6710	0.0794	0.7104	0.0714	0.6412	0.1092	**0.7380**	0.0622	0.7102	0.0218
Bupa	0.6108	0.0688	0.6491	0.0450	0.6108	0.0852	**0.6547**	0.0455	0.6468	0.0433	0.5909	0.0310	0.6345	0.0437	0.6313	0.0645	0.6404	0.0916
Cleveland	0.5314	0.0745	0.5149	0.0703	0.4592	0.1080	0.5510	0.0623	0.4919	0.0696	0.5251	0.0836	0.4916	0.0575	**0.5681**	0.0651	0.5378	0.0392
Contraceptive	0.4277	0.0369	0.4908	0.0504	**0.5323**	0.0528	0.4861	0.0441	0.4494	0.0282	0.4956	0.0452	0.4718	0.0355	0.4542	0.0351	0.4562	0.0225
Crx	0.7957	0.0512	**0.8435**	0.0477	0.8188	0.0444	0.8362	0.0534	0.8159	0.0381	0.7000	0.1612	0.6029	0.0439	0.8188	0.0417	0.8333	0.0517
Dermatology	0.9535	0.0345	0.9507	0.0444	0.9402	0.0378	0.9672	0.0282	0.9047	0.0732	0.7791	0.1695	0.9016	0.0453	0.9399	0.0458	**0.9700**	0.0240
Ecoli	0.807	0.0751	0.7797	0.0383	0.7352	0.0512	**0.8216**	0.0462	0.7621	0.0644	0.6870	0.0823	0.7086	0.0682	0.8154	0.0415	0.7888	0.0348
German	0.705	0.0425	0.6930	0.0564	0.6890	0.0357	**0.7110**	0.0370	0.5950	0.0425	0.6730	0.0250	0.6520	0.0343	0.7020	0.0140	0.7010	0.0088
Glass	**0.7361**	0.1191	0.6568	0.1286	0.6644	0.1313	0.6461	0.1110	0.6946	0.1216	0.5934	0.1292	0.7050	0.1166	0.7118	0.0768	0.6481	0.1216
Haberman	0.6697	0.0546	0.7119	0.0594	0.4937	0.0778	0.6859	0.0739	0.6405	0.0343	0.6991	0.0622	0.6796	0.0385	0.7086	0.0770	**0.7449**	0.0600
Hepatitis	0.8075	0.1109	0.7933	0.0947	0.7221	0.1418	0.8254	0.0455	**0.8388**	0.0695	0.7938	0.0238	0.7688	0.0947	0.7938	0.0238	0.7938	0.0238
Iris	0.9333	0.0516	0.9667	0.0471	0.9600	0.0344	0.9600	0.0466	0.9400	0.0492	0.9600	0.0344	0.9467	0.0422	0.9400	0.0378	**0.9667**	0.0471
Led7digit	0.402	0.0948	**0.7140**	0.0481	0.4820	0.0503	0.6020	0.0856	0.6520	0.0464	0.4700	0.0901	0.0900	0.0141	0.6860	0.0534	0.3340	0.0589
Lymphography	0.7387	0.0877	0.7427	0.1175	0.7580	0.0897	0.8006	0.0966	0.7612	0.0993	0.5826	0.1465	**0.8130**	0.0752	0.7733	0.0911	0.7577	0.0949
Mammographic	0.7368	0.0559	0.8253	0.0673	0.8033	0.0530	0.7565	0.0357	0.7721	0.0526	**0.8304**	0.0436	0.6254	0.0291	0.8002	0.0689	0.8065	0.0561
Movement	0.8194	0.0434	0.6194	0.0525	0.5833	0.0818	0.7389	0.0755	0.7500	0.0752	0.7444	0.0504	**0.8583**	0.0480	0.7417	0.0898	0.6583	0.0718
Newthyroid	**0.9723**	0.0226	0.9353	0.0657	0.9485	0.0521	0.9580	0.0269	0.9580	0.0405	0.8985	0.0793	0.9487	0.0347	0.9632	0.0360	0.9450	0.0469
Pima	0.7033	0.0353	0.7385	0.0516	0.6771	0.0454	0.7278	0.0432	0.6432	0.0598	0.7165	0.0597	0.7045	0.0409	**0.7501**	0.0363	0.7462	0.0459
Saheart	0.6449	0.0399	0.6861	0.0401	0.5736	0.0376	0.6819	0.0504	0.5692	0.0801	0.6754	0.0601	0.6170	0.0636	**0.7273**	0.0512	0.7057	0.0582
Sonar	**0.8555**	0.0751	0.7055	0.1500	0.7262	0.0783	0.6443	0.1355	0.7690	0.0871	0.8026	0.0961	0.8412	0.0716	0.7405	0.1080	0.7695	0.1121
Spectfheart	0.697	0.0655	0.7460	0.0696	0.7077	0.0662	0.7942	0.0175	**0.8091**	0.0263	0.7942	0.0175	0.7194	0.0813	0.7942	0.0175	0.7942	0.0175
Tic-tac-toe	0.7307	0.0256	0.8445	0.0512	0.9676	0.0200	0.9207	0.0264	0.8631	0.0424	0.7088	0.0329	**1.0000**	0.0000	0.9206	0.0275	0.7067	0.0234
Vehicle	**0.701**	0.056	0.6607	0.0543	0.6832	0.0428	0.6597	0.0550	0.6702	0.0491	0.5956	0.0921	0.6159	0.0500	0.6738	0.0307	0.6456	0.0458
Wine	0.9552	0.0485	0.9490	0.0619	0.9216	0.0534	**0.9660**	0.0293	0.9438	0.0524	0.8592	0.0618	0.9438	0.0524	0.9431	0.0661	0.9542	0.0538
Wisconsin	0.9557	0.0259	0.9585	0.0171	0.9499	0.0272	**0.9700**	0.0281	0.9456	0.0329	0.9285	0.0301	0.9657	0.0254	0.9613	0.0287	0.9642	0.0227
Yeast	0.5047	0.0391	0.5404	0.0397	0.5014	0.0260	0.5735	0.0394	0.5162	0.0470	0.4946	0.0555	0.5425	0.0237	0.5634	0.0316	**0.5735**	0.0346
Zoo	0.9281	0.0657	0.9281	0.0692	0.9408	0.0703	0.9683	0.0524	**0.9683**	0.0524	0.8761	0.0664	0.9617	0.0685	0.9500	0.0671	0.9131	0.0802
Average	0.7458		0.7608		0.7228		0.7689		0.7389		0.7214		0.7247		0.7750		0.7531	

Table 4
Average number of rules yielded over small data sets.

Data sets	C4.5Rules		RIPPER		BNGE		RISE		INNER		SIA		EHS-CHC		Filtered EHS-CHC	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Appendicitis	3.20	0.42	7.50	1.27	26.80	3.85	52.40	4.50	8.50	1.90	46.40	4.40	9.90	0.74	**2.00**	0.00
Australian	11.60	3.72	25.70	2.31	239.20	13.60	226.00	5.72	**4.00**	1.05	599.50	3.72	30.20	2.82	10.30	1.06
Balance	34.70	4.03	37.60	5.85	342.50	12.80	537.10	13.30	**9.70**	1.64	112.80	3.97	13.40	1.71	10.60	0.97
Breast	12.40	2.07	36.50	4.06	82.60	3.03	208.70	9.04	9.10	2.73	95.10	4.09	28.20	1.99	**6.20**	1.03
Bupa	**8.10**	1.20	22.60	2.91	183.20	13.04	201.50	4.65	33.10	7.43	220.30	4.74	44.30	2.41	26.20	1.93
Cleveland	10.70	1.25	44.10	4.41	130.50	4.25	161.70	4.76	13.10	2.51	252.30	8.81	18.90	3.00	**5.20**	1.03
Contraceptive	30.80	1.87	64.50	7.21	1211.80	14.51	1018.40	15.88	**8.40**	1.58	530.40	4.70	55.60	8.83	29.70	2.50
Crx	15.50	2.59	25.00	3.46	109.10	7.71	286.90	11.32	26.10	12.11	579.50	7.53	34.40	2.80	**15.00**	2.54
Dermatology	**9.10**	0.32	15.00	1.33	25.00	5.29	74.50	17.81	38.70	8.77	59.90	3.87	12.30	2.06	10.10	1.37
Ecoli	**11.70**	1.34	35.00	3.65	93.10	6.87	169.20	11.11	15.50	4.28	194.50	9.19	29.90	2.18	17.30	1.64
German	26.80	2.15	37.10	4.01	232.90	7.95	676.60	13.87	50.20	2.49	567.40	9.67	38.60	4.99	**18.70**	3.16
Glass	**10.50**	1.65	22.80	2.44	78.60	6.02	92.20	4.49	18.10	2.47	187.80	2.90	27.60	1.58	13.50	1.72
Haberman	**4.30**	0.82	18.90	2.33	212.20	7.27	132.40	2.88	17.80	4.80	119.00	5.16	26.60	1.96	8.10	1.20
Hepatitis	6.30	1.42	7.90	1.73	37.90	2.81	50.80	3.49	14.70	1.95	69.40	6.90	7.50	1.35	**4.40**	1.26
Iris	5.00	0.00	6.20	0.79	12.20	1.32	37.10	11.21	8.20	1.23	13.90	1.79	6.20	0.79	**5.00**	0.00
Led7digit	16.10	1.29	92.50	4.17	403.50	4.58	279.00	34.57	10.30	1.89	**1.00**	0.00	12.90	1.10	5.00	0.94
Lymphography	11.50	1.18	13.60	1.65	29.20	2.35	81.00	10.35	8.30	1.83	33.30	1.70	13.40	1.35	**7.70**	1.57
Mammographic	7.70	1.06	25.40	6.57	754.40	10.44	315.70	11.90	**7.10**	1.10	188.00	4.27	22.60	2.22	9.60	2.07
Movement	**27.30**	2.79	51.80	3.79	83.90	5.95	130.60	7.50	71.80	1.87	299.70	2.63	45.70	4.08	35.30	2.95
Newthyroid	6.80	0.42	7.00	0.67	18.60	3.41	33.90	5.67	7.30	1.49	40.80	8.64	7.20	1.14	**5.00**	0.47
Pima	**8.80**	2.30	24.70	3.27	339.60	8.06	436.90	7.40	13.00	5.83	600.80	20.50	50.10	8.25	29.00	2.31
Saheart	**6.70**	1.42	24.00	3.77	178.40	2.41	260.90	12.56	14.40	7.56	415.10	0.99	63.50	9.62	20.40	3.03
Sonar	8.70	1.16	**8.40**	0.70	58.80	9.67	59.10	2.73	40.60	2.01	187.20	0.42	26.30	2.71	20.00	1.70
Spectfheart	11.20	1.55	10.00	1.33	62.10	1.66	168.70	23.53	45.10	4.01	240.30	0.48	20.80	2.20	**7.60**	2.46
Tic-tac-toe	48.60	8.97	16.20	1.23	124.90	21.69	455.60	16.41	56.60	4.20	31.20	3.01	16.70	1.16	**10.10**	1.20
Vehicle	**19.10**	2.13	45.60	6.29	349.60	6.64	316.50	14.21	108.20	8.53	680.50	14.56	75.70	9.72	46.10	3.54
Wine	**5.00**	0.00	5.60	1.07	10.10	0.99	29.70	10.78	6.20	1.03	160.20	0.42	9.80	1.81	7.60	0.97
Wisconsin	10.00	2.11	9.80	1.40	64.30	5.36	182.90	87.06	7.80	3.19	41.30	2.75	8.40	1.17	**4.40**	1.07
Yeast	33.00	3.46	139.60	6.79	902.20	14.88	1067.70	15.90	**24.20**	4.26	943.30	21.91	102.00	21.17	38.40	3.57
Zoo	8.70	0.48	8.70	0.48	9.00	0.47	25.60	3.75	8.90	0.88	10.10	0.74	7.00	0.00	**6.30**	0.67
Average	14.33		29.64		213.54		258.98		23.50		250.70		28.86		14.49	

Observing Tables 3 and 4, we can make the following analysis:

- The *EHS-CHC* proposal obtains the best average result in accuracy over test data. It clearly outperforms all the other techniques. However, *filtered EHS-CHC* cannot improve the accuracy obtained by BNGE and C4.5Rules.
- The best classical approaches for small data sets are 1NN and C4.5Rules, whereas BNGE and *EHS-CHC* are the most remarkable NGE based approaches in accuracy.
- It seems that there is no general rule that could predict which algorithm is the best choice given a certain data set. All techniques achieved, at least once, the best accuracy result in different data sets.
- Except for C4.5Rules, *filtered EHS-CHC* produces hyperrectangle models with a lower number of hyperrectangles than the remaining methods. On the other hand, *EHS-CHC* requires more hyperrectangles to improve the performance in accuracy. The INNER approach, together with C4.5Rules are the techniques that produce the lowest number of rules.
- The trade-off between accuracy and simplicity is maintained by both *EHS-CHC* and *filtered EHS-CHC*. Other NGE learners, such as BNGE, RISE and SIA, need to store a large number of hyperrectangles to obtain good performances, but our approach is able to obtain similar results with a much lower set of hyperrectangles.

Clearly, the best approach could be chosen depending on the interests of the users: looking for more precision or more simplicity. The *EHS-CHC* approach fits the first purpose and *filtered EHS-CHC* fits the second. Nevertheless, we want to illustrate that both proposals have similar behavior when statistical analysis is conducted. Table 5 illustrates a Wilcoxon comparison between them in accuracy. No significant differences are detected, but *EHS-CHC* collects a higher value of ranking. In small data sets, the number of rules needed to build a model is, in general, not very high and we could establish that the accuracy is the preferred objective in these domains. Thus, in the rest of the study of small data sets, we will use the non filtered *EHS-CHC* version to compare with other techniques.

Table 6 collects the results of applying the Wilcoxon test to *EHS-CHC* and the rest of the algorithms studied in this paper considering small data sets. This table is composed of two columns: In the first one, the measure of performance used is the accuracy classification in test set and in the second one, we carry out the Wilcoxon test by using as a performance measure the number of hyperrectangles/rules yielded by the techniques. The table is composed of N_a rows where N_a is the number of algorithms considered in this study. In each one of the cells, three symbols can appear: +, = or −. They represent that *EHS-CHC* outperforms (+), is similar (=) or is worse (−) in performance than *EHS-CHC*. The value in brackets is the p-value obtained in the comparison and the level of significance considered is $\alpha = 0.10$.

Other statistical studies can be performed by using non-parametric multiple comparison tests. These types of procedures consider the set of results obtained by all the algorithms to compute a ranking which represents the performance of the associated algorithms. The smaller the ranking, the better the algorithm. The three multiple comparison tests described in Section 4.3 are used and the rankings are depicted in Table 7.

The statistical support provided allows us to conclude the following:

- *EHS-CHC* is the best approach compared with the rest of techniques. As we can see, the Wilcoxon test confirms that it outperforms all of them in terms of accuracy, except for the BNGE technique, which behaves similarly to our approach. However, BNGE requires a lot of hyperrectangles to achieve this accuracy rate.
- Considering multiple comparison tests, the difference in ranking between BNGE and *EHS-CHC* is rather small. In fact, in Friedman and Friedman Aligned Ranks tests, BNGE has a lower ranking than *EHS-CHC*. However, the opposite is true in the Quade test. This latter test positively weights more difficult data sets when computing the ranking, and the difficulty is measured depending on the differences of performance registered by all methods. This could suggest that *EHS-CHC* is better suited to more complicated problems.

Finally, Fig. A.1 in Appendix A illustrates the comparison of *EHS-CHC* with the remaining techniques considered in this study in terms of accuracy over test data.

5.2. Results and analysis for medium data sets

Table 8 shows the results measured by accuracy in test data for each approach considered in this paper. Table 9 shows the average number of rules/hyperrectangles yielded by the approaches considered. The same format followed in the previous section is used. SIA and RISE algorithms could not be run over medium data sets for efficiency reasons.

Observing Tables 8 and 9, we can point out the following analysis:

- Both evolutionary proposals obtain the best average result in accuracy over test data in medium data sets. *Filtered EHS-CHC* is even better than *EHS-CHC*.
- 1NN is better suited for medium data sets in accuracy results than classical NGE techniques, although it is outperformed by the evolutionary approaches. Nevertheless, it should be noted that this method does not give interpretable results.

Table 5
Wilcoxon comparison of the proposed models in accuracy over small data sets.

	R^+	R^-	p-value
EHS-CHC vs. filtered EHS-CHC	284.5	180.5	0.255

Table 6
Wilcoxon test results in accuracy and number of rules considering small data sets.

Algorithm	EHS-CHC	
	Accuracy	num. rules
1NN	+ (0.028)	–
C4.5Rules	+ (0.045)	– (0.01)
RIPPER	+ (0.001)	= (0.746)
BNGE	= (0.964)	+ (0.000)
RISE	+ (0.005)	+ (0.000)
INNER	+ (0.001)	= (0.221)
SIA	+ (0.019)	+ (0.000)

Table 7
Ranking values obtained by the multiple comparisons non-parametric tests over small data sets.

	1NN	C4.5Rules	RIPPER	BNGE	RISE	INNER	SIA	EHS-CHC
Friedman	4.77	3.98	5.47	**3.13**	5.00	5.43	4.88	3.33
Friedman Aligned Ranks	123.10	108.88	150.55	**81.90**	133.50	145.32	138.32	82.43
Quade	4.66	3.94	5.47	3.45	4.99	5.27	4.90	**3.33**

Table 8
Accuracy results in test over medium data sets.

Data sets	1NN		C4.5Rules		RIPPER		BNGE		INNER		EHS-CHC		Filtered EHS-CHC	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Abalone	0.1991	0.0160	0.2262	0.0163	0.2346	0.0183	0.2240	0.0244	0.2149	0.0310	0.2173	0.0106	**0.2640**	0.0108
Banana	0.8751	0.0103	0.8836	0.0154	0.6489	0.0488	0.8855	0.0168	0.6366	0.0638	**0.8975**	0.0108	0.8968	0.0072
Coil2000	0.8963	0.0077	0.9330	0.0041	0.9322	0.0050	0.9308	0.0039	0.8737	0.2108	**0.9403**	0.0005	0.9401	0.0009
Magic	0.8059	0.0090	0.8201	0.0134	**0.8340**	0.0217	0.8232	0.0079	0.7550	0.0403	0.8141	0.0078	0.8132	0.0089
Page-blocks	0.9576	0.0101	0.9587	0.0123	0.9622	0.0085	**0.9624**	0.0061	0.9280	0.0119	0.9457	0.0103	0.9494	0.0075
Penbased	**0.9935**	0.0023	0.9512	0.0080	0.9616	0.0057	0.9643	0.0052	0.5425	0.0490	0.9741	0.0045	0.9727	0.0040
Phoneme	**0.8991**	0.0175	0.8229	0.0155	0.8066	0.0233	0.8662	0.0180	0.7554	0.0376	0.8407	0.0219	0.8107	0.0202
Satimage	**0.9058**	0.0132	0.8345	0.0156	0.8559	0.0136	0.8701	0.0142	0.7448	0.0257	0.8625	0.0119	0.8696	0.0075
Segment	**0.9662**	0.0070	0.9545	0.0094	0.9541	0.0118	0.9437	0.0189	0.8563	0.0253	0.9364	0.0119	0.9355	0.0152
Splice	0.7495	0.0115	0.9223	0.0133	**0.9348**	0.0160	0.7542	0.0282	0.7085	0.0293	0.9251	0.0137	0.9298	0.0132
Texture	**0.9905**	0.0041	0.9042	0.0158	0.9280	0.0060	0.9400	0.0104	0.6491	0.0371	0.9505	0.0083	0.9489	0.0060
Twonorm	**0.9468**	0.0073	0.8642	0.0101	0.9107	0.0136	0.9318	0.0244	0.7565	0.0387	0.9173	0.0122	0.9182	0.0125
Average	0.8488		0.8396		0.8303		0.8413		0.7018		0.8518		0.8541	

Table 9
Average number of rules yielded over medium data sets.

Data sets	C4.5Rules		RIPPER		BNGE		INNER		EHS-CHC		Filtered EHS-CHC	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Abalone	125.70	4.16	305.80	11.61	3174.10	24.10	63.80	4.57	1874.70	12.82	**26.70**	1.64
Banana	40.30	6.38	15.00	2.31	3573.40	24.28	**7.60**	2.59	178.60	40.49	67.30	8.33
Coil2000	42.20	1.93	19.80	1.81	2648.30	46.10	28.70	8.82	137.40	39.95	**10.70**	2.16
Magic	67.00	20.57	74.40	11.27	9010.10	150.02	**11.50**	4.06	2514.00	300.89	379.50	139.77
Page-blocks	**20.30**	2.21	50.40	4.33	557.70	19.01	26.00	4.22	53.40	7.23	33.10	2.69
Penbased	128.60	8.63	108.70	4.72	1621.60	60.73	**37.90**	6.08	339.40	48.44	299.10	40.99
Phoneme	30.10	6.61	54.60	5.82	2706.60	27.36	**16.00**	11.02	531.00	87.13	221.40	59.60
Satimage	73.20	13.30	108.80	4.29	1349.80	28.98	**56.50**	13.03	395.60	64.19	216.80	48.03
Segment	**27.30**	2.91	34.60	3.06	236.20	22.18	57.60	13.44	80.80	7.66	60.00	8.35
Splice	145.00	29.86	38.30	4.81	578.00	20.58	78.40	3.72	25.60	3.17	**18.00**	2.62
Texture	86.00	12.73	**83.10**	3.67	798.60	19.53	126.50	27.35	328.60	44.09	278.90	32.44
Twonorm	47.80	9.87	57.50	2.72	2128.30	25.38	**8.70**	2.71	57.40	11.79	58.80	10.63
Average	69.46		79.25		2365.23		**43.27**		543.04		139.19	

Table 10
Wilcoxon comparison of the proposed models in accuracy over medium data sets.

	R^+	R^-	p-value
EHS-CHC vs. filtered EHS-CHC	35	43	0.753

Table 11
Wilcoxon test results in accuracy and number of rules considering medium data sets.

Algorithm	Filtered EHS-CHC	
	Accuracy	num. rules
1NN	= (0.814)	–
C4.5Rules	+ (0.084)	– (0.084)
RIPPER	= (0.272)	= (0.136)
BNGE	= (0.875)	+ (0.002)
INNER	+ (0.002)	– (0.050)

- All classical approaches behave well, whereas BNGE and *EHS-CHC* are the most remarkable NGE based approaches in accuracy. Note that the RIPPER algorithm is more competitive in larger data sets rather than smaller.
- *Filtered EHS-CHC* produces models with a lower number of hyperrectangles than other NGE schemes, except INNER. On the other hand, *EHS-CHC* uses more hyperrectangles without obtaining a significant improvement in accuracy with respect to the filtered version.
- C4.5Rules and RIPPER obtain very interpretable models, but 1NN outperforms them in accuracy because it can represent more complex decision surfaces. This fact supports the NGE theory, but it is very difficult to optimize a set of hyperrectangles in large domains, such as in medium size data sets.
- Some NGE methods cannot be run over medium data sets: RISE and SIA. Furthermore, BNGE needs a lot of hyperrectangles and it cannot outperform 1NN. Evolutionary based NGE methods based on hyperrectangles' selection have been shown to be very useful for larger problems, improving the results obtained by classical rule learners and 1NN.

Table 10 illustrates a Wilcoxon comparison between the *EHS-CHC* and *filtered EHS-CHC* approaches in accuracy over medium data sets. Again no significant differences are detected, but *filtered EHS-CHC* collects a higher value of ranking. Now, there is no doubt about which approach is the best. *Filtered EHS-CHC* outperforms *EHS-CHC* in accuracy requiring less hyperrectangles when dealing with medium data sets.

Table 11 collects the results of applying the Wilcoxon test to *Filtered EHS-CHC* and the rest of algorithms studied in this paper considering small data sets. The format was explained in the previous section.

Again, the three multiple comparison tests are used to compute the rankings, which are presented in Table 12.

The statistical support provided allows us to conclude the following:

- *Filtered EHS-CHC* is better than C4.5Rules and INNER. As we can see, the Wilcoxon test confirms that it outperforms them in accuracy, but its results are competitive with those obtained by 1NN, RIPPER and BNGE. However, BNGE requires many hyperrectangles to achieve this accuracy rate, and 1NN do not perform any reduction at all.
- Results offered by RIPPER and *filtered EHS-CHC* are similar in both objectives.
- The rankings computed by multiple comparison procedures are again very close for BNGE and our approach. With Quade test, 1NN obtains the best ranking, indicating to us that more complex problems require more complex decision modeling.

Table 13
Orders of efficiency in the worst case for each NGE model.

NGE model	Time complexity
SIA	$O(2^M \cdot n^3 \cdot C)$
BNGE	$O(n^3 \cdot M)$
RISE	$O(n^3 \cdot M^2)$
INNER	$O(n^2 \cdot M^2) + O(n \cdot M^3)$
EHS-CHC	$O(E \cdot n^3 \cdot M)$

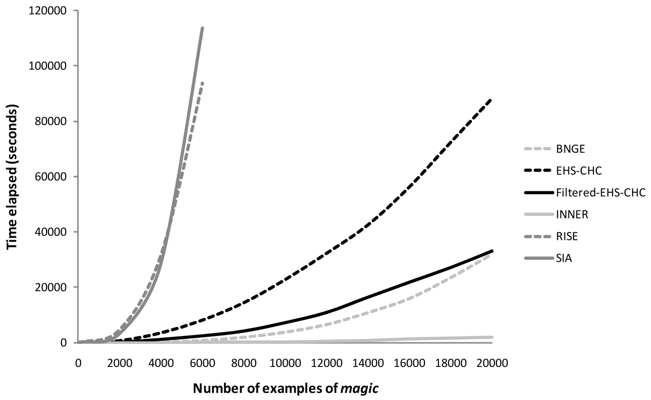


Fig. 3. Computational times considering subsamples of *magic* data set.

Fig. A.2 in Appendix A illustrates the comparison of *filtered EHS-CHC* with the remaining techniques considered in this study in terms of accuracy over test data. Star plots are used for a clear presentation of differences among the methods studied for each data set.

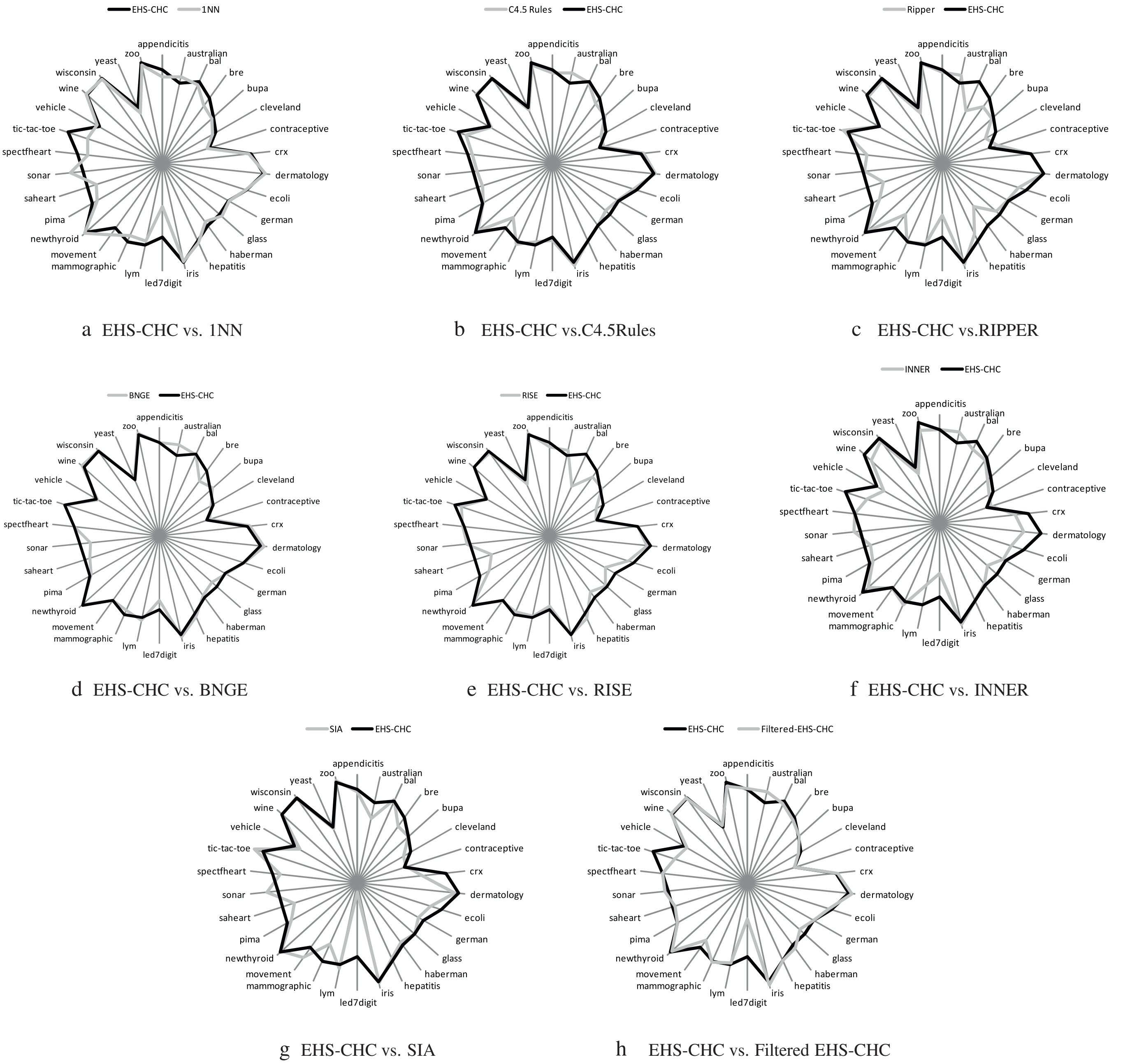
5.3. Study of running time of NGE models

In this subsection, we study the time consumption of the NGE models considered in this paper. We will proceed in two steps starting with the estimation order of the theoretical time complexity in the improbable worst case. We then report a figure with CPU times obtained from random samples of the *magic* data set.

Table 13 shows the theoretical time complexity of the NGE models in the worst case, where n denotes the number of examples in the data set, M denotes the number of features and E denotes the number of evaluations performed by the EA considered and C is the number of classes. Fig. 3 illustrates the computational time obtained by each model considering subsamples of *magic* data set. The experiment performed considers the average results of 10 runs on each subset. We take the data set *magic* and add artificial examples up to 20,000 samples. Then, we perform random partitions of the complete set from 2000 samples to 18,000 samples. This

Table 12
Ranking values obtained by the multiple comparisons non-parametric tests over medium data sets.

	1NN	C4.5Rules	RIPPER	BNGE	INNER	Filtered EHS-CHC
Friedman	2.92	3.50	3.17	**2.75**	5.92	**2.75**
Friedman Aligned Ranks	29.33	34.75	33.25	30.17	64.17	**27.33**
Quade	**2.37**	3.87	3.53	2.77	5.97	2.49



a EHS-CHC vs. 1NN

b EHS-CHC vs.C4.5Rules

c EHS-CHC vs.RIPPER

d EHS-CHC vs. BNGE

e EHS-CHC vs. RISE

f EHS-CHC vs. INNER

g EHS-CHC vs. SIA

h EHS-CHC vs. Filtered EHS-CHC

Fig. A.1. Accuracy in small data sets..

methodology allows us to compute the degree of time complexity varying the number of instances and keeping constant the number of features.[3]

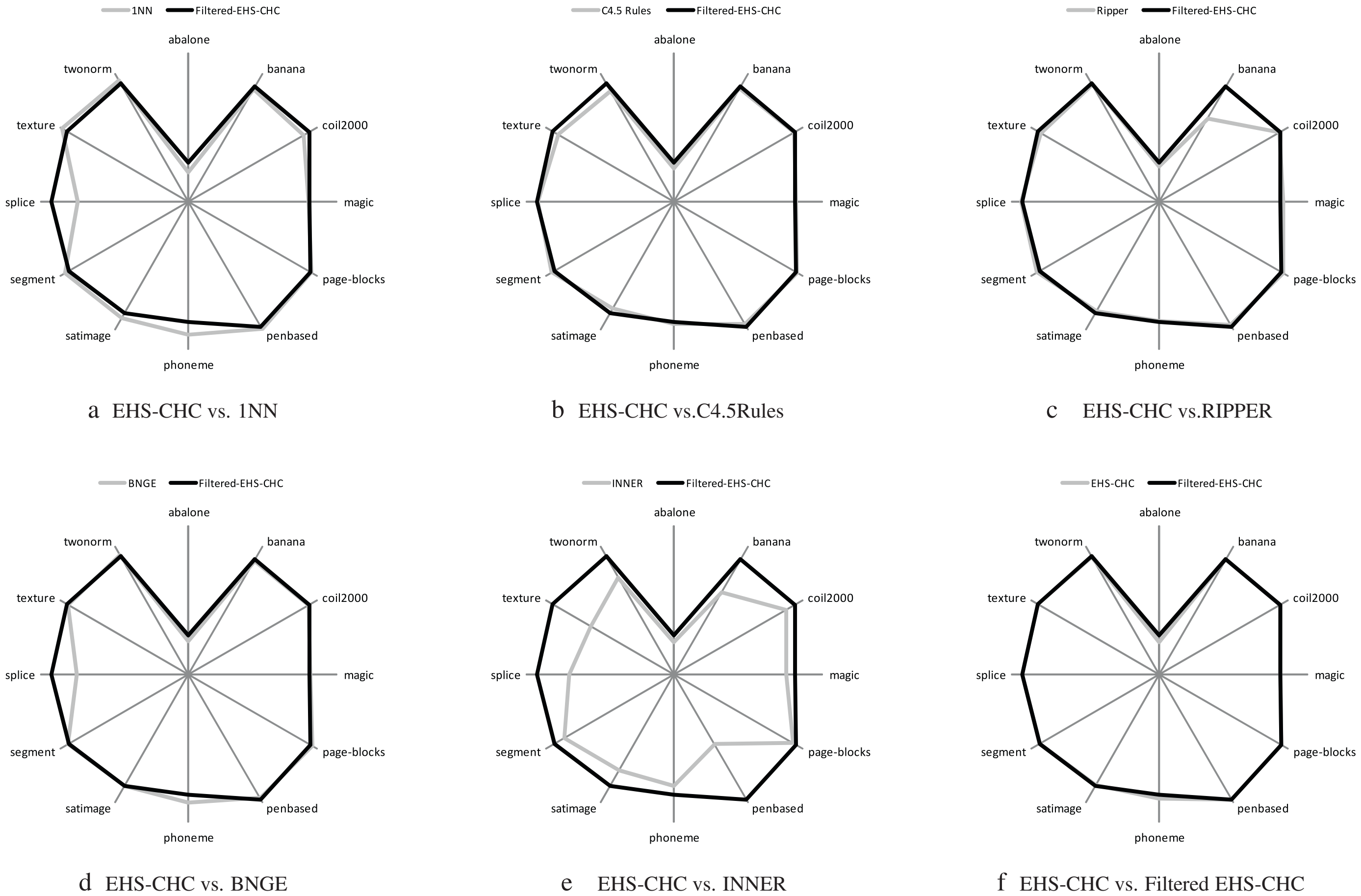
Taking into account both theoretical and empirical perspective, we can highlight the following about the time complexity study:

- SIA and RISE are very slow and they cannot handle data sets with more than 6000 examples in a reasonable time.
- INNER has the best time complexity and hence the slope of the associated curve associated is much lower and it may allow us to use this algorithm over huge data sets.
- *Filtered EHS-CHC* is more efficient than *EHS-CHC* because the filter process reduces both the number of hyperrectangles and the complexity of them, reducing also the computations of distances.
- When *n* increases, *filtered EHS-CHC* may require lower time computation than BNGE in a certain moment, due to the fact that the number of evaluations and other operations carried out by it suppose a multiplicative constant of less magnitude. The graphic would show a cross between both curves if n continue to rise.

6. Concluding remarks

The purpose of this paper is to present a proposal of an evolutionary hyperrectangle selection algorithm for nested generalized exemplar learning in classification called *EHS-CHC*. It creates an initial set of hyperrectangles from training data and then it performs a selection process focused on maximizing the accuracy and coverage of examples with the lowest possible number of hyperrectangles.

[3] The machine used was an Intel Core i7 CPU 920 at 2.67 GHz with 4GB of RAM.



a EHS-CHC vs. 1NN

b EHS-CHC vs.C4.5Rules

c EHS-CHC vs.RIPPER

d EHS-CHC vs. BNGE

e EHS-CHC vs. INNER

f EHS-CHC vs. Filtered EHS-CHC

Fig. A.2. Accuracy in medium data sets.

The results show that *EHS-CHC* allows us to obtain very accurate models with a low number of hyperrectangles. We have compared it with several classical and advanced NGE learning approaches and the effectiveness of the models obtained is very competitive with respect to them. In larger data sets, the improvement achieved is even more remarkable, improving the results obtained by classical rule learners and 1NN. Future work in this topic could be focused on the hybridization of evolutionary models for clustering [36] with *EHS-CHC* to obtain a more accurate set of initial hyperrectangles to be selected.

Acknowledgement

This work was supported by TIN2008-06681-C06-01. J. Derrac holds a research scholarship from the University of Granada. J. Luengo holds a FPU scholarship from from Spanish Ministry of Education and Science. We are grateful to the reviewers for identifying essential issues and providing us with important and valuable feedback. We have found the suggestions and comments in the reviews very useful to improve the quality of this paper.

Appendix A. Star plots in small and medium data sets

In this appendix, we include the star plot representations in accuracy for the comparison between the best proposed algorithm in small and medium data sets with the remaining methods. These star plots represent the performance as the distance from the center; hence a higher area determines the best average performance. The plots allow us to visualize the performance of the algorithms comparatively for each problem and in general. Figs. A.1 and A.2 illustrate the star plots for small and medium size data sets respectively.

References

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

[2] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2005.

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

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

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

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

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

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

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

[10] J. Ranilla, A. Bahamonde, FAN. Finding accurate inductions, International Journal of Human Computer Studies 56 (2002) 445–474.

[11] D.G. Heath, S. Kasif, S.R. Kosaraju, S. Salzberg, G.F. Sullivan, Learning nested concept classes with limited storage, Journal of Experimental and Theoreticall Artificial Intelligence 8 (2) (1996) 129–147.

[12] L.B. Figueira, M. do Carmo Nicoletti, Evaluating the effects of distance metrics on a NGE-based system, in: Proceedings of the I.E.E.E. SMC, vol. 4, 2004, pp. 3395–3401.

[13] D.R. Wilson, T.R. Martinez, Improved heterogeneous distance functions, Journal of Artificial Intelligence Research 6 (1997) 1–34.

[14] F.O.S. de Sá Lisboa, M. do Carmo Nicoletti, A. Ramer, A version of the nge model suitable for fuzzy domains, Journal of Intelligent Fuzzy Systems 18 (1) (2007) 1–17.

[15] M. Cintra, H. Camargo, E. Hruschka, M. do Carmo Nicoletti, Automatic construction of fuzzy rule bases: a further investigation into two alternative inductive approaches, Journal of Universal Computer Science 14 (15) (2008) 2456–2470.

[16] M. do Carmo Nicoletti, L.B. Figueira, E.R. Hruschka Jr., Transferring neural network based knowledge into an exemplar-based learner, Neural Computing and Applications 16 (3) (2007) 257–265.

[17] M.J.R.B. do Carmo Nicoletti Jr., Constructive neural network algorithms for feedforward architectures suitable for classification tasks, in: D.A. Elizondo, L. Franco, J.M. Jerez (Eds.), Constructive Neural Networks, Vol. 258 of Studies in Computational Intelligence, Springer, 2009, p. 23.

[18] C.A. Policastro, A.C. Carvalho, A.C. Delbem, A hybrid case adaptation approach for case-based reasoning, Applied Intelligence 28 (2) (2008) 101–119.

[19] D.R. Wilson, T.R. Martinez, Reduction techniques for instance-based learning algorithms, Machine Learning 38 (3) (2000) 257–286.

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

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

[22] A. Zafra, E.L. Gibaja, S. Ventura, Multiple instance learning with multiple objective genetic programming for web mining, Applied Soft Computing 11 (1) (2011) 93–102.

[23] D. Whitley, R. Beveridge, C. Guerra, C. Graves, Messy genetic algorithms for subset feature selection, in: Proceedings of the International Conference on Genetic Algorithms, 1998, pp. 568–575.

[24] C. Guerra-Salcedo, S. Chen, D. Whitley, S. Smith, Fast and accurate feature selection using hybrid genetic strategies, CEC (1999) 177–184.

[25] X. Wang, J. Yang, X. Teng, W. Xia, R. Jensen, Feature selection based on rough sets and particle swarm optimization, Pattern Recognition Letters 28 (4) (2007) 459–471.

[26] H. Yan, J. Zheng, Y. Jiang, C. Peng, S. Xiao, Selecting critical clinical features for heart diseases diagnosis with a real-coded genetic algorithm, Applied Soft Computing 8 (2008) 1105–1111.

[27] M. Marinaki, Y. Marinakis, C. Zopounidis, Honey bees mating optimization algorithm for financial classification problems, Applied Soft Computing 10 (3) (2010) 806–812.

[28] 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.

[29] J.R. Cano, F. Herrera, M. Lozano, Evolutionary stratified training set selection for extracting classification rules with trade-off precision-interpretability, Data and Knowledge Engineering 60 (2007) 90–108.

[30] 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.

[31] I. Turkoglu, E.D. Kaymaz, A hybrid method based on artificial immune system and k-nn algorithm for better prediction of protein cellular localization sites, Applied Soft Computing 9 (2009) 497–502.

[32] H. Ahn, K.J. Kim, Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach, Applied Soft Computing 9 (2009) 599–607.

[33] 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.

[34] 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.

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

[36] 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.

[37] G.S.I.A. Venturini, A supervised inductive algorithm with genetic search for learning attributes based concepts, ECML (1993) 280–296.

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

[39] 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.

[40] 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.

[41] 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.

[42] S. García, J. Derrac, J. Luengo, F. Herrera, A first approach to nearest hyperrectangle selection by evolutionary algorithms, in: ISDA '09 Proceedings of the 2009 Ninth International Conference on Intelligent Systems Design and Applications, 2009, pp. 517–522.

[43] A. Orriols-Puig, J. Casillas, E. Bernadó-Mansilla, Genetic-based machine learning systems are competitive for pattern recognition, Evolutionary Intelligence 1 (3) (2008) 209–232.

[44] 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.

[45] D.L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Transactions on Systems, Man, and Cybernetics 2 (1972) 408–421.

[46] A. Asuncion, D. Newman, UCI machine learning repository, 2007, URL: http://www.ics.uci.edu/mlearn/MLRepository.html.

[47] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Computing 13 (3) (2008) 307–318.

[48] 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.

[49] D. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, Chapman & Hall/CRC, 2006.