Available online at www.sciencedirect.com
ScienceDirect
Data & Knowledge Engineering 60 (2007) 90–108
DATA & KNOWLEDGE ENGINEERING
www.elsevier.com/locate/datak

# Evolutionary stratified training set selection for extracting classification rules with trade off precision-interpretability ☆

José Ramón Cano [a,*], Francisco Herrera [b], Manuel Lozano [b]

[a] Department of Computer Science, University of Jaén, 23700 Linares, Jaén, Spain
[b] Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain

Available online 3 March 2006

**Abstract**

The generation of predictive models is a frequent task in data mining with the objective of generating highly precise and interpretable models. The data reduction is an interesting preprocessing approach that can allow us to obtain predictive models with these characteristics in large size data sets. In this paper, we analyze the rule classification model based on decision trees using a training selected set via evolutionary stratified instance selection. This method faces the scaling problem that appears in the evaluation of large size data sets, and the trade off interpretability-precision of the generated models.


*Keywords:* Training set selection; Interpretability; Precision; Evolutionary algorithms; Rule classification; Decision trees

## 1. Introduction

A basic process in data mining is the generation of representative models from data [1]. The models, depending on their domain of application, can be descriptive or predictive. The classical objective of predictive models is the accuracy or precision of the model. On the other hand, the interpretability of the model is an important aspect for the expert point of view, to understand the model behaviour [2]. In classical literature, we can find different proposals to measure the quality of the predictive models, as well as the precision, like simplicity, interpretability, etc. [3].

In this paper we are going to focus our attention on the predictive models based on classification rules for different size data sets, with the special interest in the trade off interpretability-precision [2]. Our models have been extracted from the data sets by means of *C4.5* algorithm [4].

☆ This work was supported by projects TIC2002-04036-C05-01 and TIN2005-08386-C05-01.
* Corresponding author.
*E-mail addresses:* jrcano@ujaen.es (J.R. Cano), herrera@decsai.ugr.es (F. Herrera), lozano@decsai.ugr.es (M. Lozano).

0169-023X/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2006.01.008

A possible way to improve the behaviour of predictive models, precision and interpretability, is to extract them from suitable reduced/selected training sets [5]. Training set selection can be developed using instance selection algorithms. The instance selection algorithms select representative instance subsets following a determined strategy, and they can improve the nearest neighbour rule prediction capabilities used in some cases as selection strategy objective [6,7]. In [5], Sebban et al. study the effect of the learning set size in decision trees performances. An important conclusion of this analysis is that the application of instance selection algorithms (and concretely, the *PSRCG* algorithm) can improve the generalization accuracy, reduce the decision tree size and tolerate the presence of noise, establishing a close link between instance selection and tree simplification.

Evolutionary algorithms (*EAs*) are adaptable methods based on natural evolution that can be applied to search and optimization problems [8–10]. The *EAs* offer interesting results when they are assessed on instance selection [11,12]. In this study, we use *CHC* algorithm as *EA* [13], considering its behaviour shown in [14]. The basic idea consists of combining in the fitness function both objectives, interpretability and precision [14,15].

The evaluation of instance selection algorithms over large size data sets makes them ineffective and inefficient. The effect produced by the size of data set in the algorithms is called scaling problem.

We focus our attention on evolutionary instance selection for large size data sets with the aim of extracting high precise-interpretable rules. To tackle the scaling problem we combine the stratification of the data sets with the instance selection over them [15]. The stratification reduces the original data set size, splitting it into strata where the selection will be applied. We analyze the selected training sets quality by means of the models (decision trees) extracted from them by means of *C4.5*, from the precision and interpretability perspectives. To compare the results we provide a statistical analysis using some statistical tests (ANOVA, Levene and Tamhane [16]).

The outline of the document is the following. In Section 2, we analyze the predictive models and their extraction using *C4.5*, presenting the measures considered to assess their behaviour. Section 3 describes the training set selection process and the drawbacks that the evaluation of very large data sets introduced in the instance selection algorithms. Section 4 presents the evolutionary stratified instance selection process applied to training set selection. Section 5 contains the experimental study developed, offering the methodology followed, the results and their analysis. Finally, in Section 6 we will point out some concluding results.

## 2. Predictive models: classification trees extraction with C4.5

The importance of decision trees and rules is that they are favoured techniques to build understandable models, a key point for the helpfulness of them and their application. A decision tree is a predictive model that can be viewed as a tree.

In this study we are going to extract the decision trees using the *C4.5* algorithm [4]. The models generated are complete and consistent, covering all the examples of the training set. The induction algorithm may over fit outliers, mislabelled, noisy data resulting in the inference of more structures than is justified by the training set. This situation is increased when the size of the learning set is large, so decision trees size is increased considerably [17–19]. The high size of the decision tree produces:

- Over fitting. In this case, the learned hypothesis is so closely related to the training examples that its generalization capabilities would be penalized [20].
- Low human interpretability. The highest size of the decision tree introduces the disadvantage of excessive complexity that can render it incomprehensible to experts [3,21].

To avoid this situation, there are several ways to simplify the decision tree, which were classified by Breslow and Aha in [22].

Among them, prune methods are more popular than the rest to be applied to the decision trees generated [23]. Prune methods can be classified in:

- Preprune methods. The prune process is developed during the tree generation. The prune determines the stopping condition for the branch specialization.
- Postprune methods. In this case, the prune process is applied after the tree construction. The prune removes nodes from bottom to top until a determined limit is reached.

Prune methods increase the generalization capabilities of the model and reduce its size, which increases its interpretability.

The drawback for both prune methods, preprune and postprune, is to determine the stop limit. The limit will depend on the training set where the decision tree is being extracted. The proper adjust of the limit produces models with better or worse behaviour. If the prune is minimal, the over fitting will be maintained. If the prune is maximal, the precision capability could be reduced due to excessive generalization.

In the case of *C4.5* algorithm, the Error-Based Pruning is applied [4]. This prune strategy has shown its balance among precision and size in decision trees generated among other sort of pruning, like Reduced Error Pruning, Pessimistic Error Pruning, Minimum Error Pruning, Critical Value Pruning, Cost-Complexity Pruning, etc. [23].

As alternative strategy to simplify the decision trees, it can be developed the reduction of the initial size of the learning set. This reduction consists of removing irrelevant instances before the induction process, often resulting in smaller trees [19,24]. This process is carried out by means of instance selection algorithms [6,7] where, instead of removing the irrelevant instances, the most representative ones are selected.

When the decision tree is going to be applied in domains where its character predictive and descriptive is important, the simplicity of the decision tree is a key factor [2]. The measures we are going to use to assess the predictive models extracted with *C4.5* will be the following [3]:

- **Test accuracy.** In predictive models learning, it is a key factor to maximize the accuracy of the set of rules obtained. This is going to be a quality measure of the model. The model will be generated by means of the *C4.5* algorithm using the training set selected. The test accuracy is calculated using the model constructed.

$$TEST = Test\ Accuracy. \tag{1}$$

- **Decision tree size.** The measure of the size of decision tree is assessed considering the number of rules ($n_R$) which compose the model.

$$SIZE = n_R. \tag{2}$$

- **Number of antecedents.** As second measure of decision tree size we introduce the mean number of antecedents per rule. Considering the rule $R_i$ as $Cond \rightarrow Class$, $N_{Antec}(R_i)$ is the number of antecedents of the rule $R_i$ and $ANT$ the mean number of antecedents in the model (see (3) and (4)):

$$N_{Antec}(R_i) = \sharp|Cond|, \tag{3}$$

$$ANT = \frac{1}{n_R}\sum_{i=1}^{n_R} N_{Antec}(R_i). \tag{4}$$

As the number of rules as the mean number of antecedents will be used to analyze the interpretability capacities of the model.

## 3. Training set selection in large size data sets

In this section the training set selection process is described. It is developed by means of instance selection algorithms, which select the most representative instances in the initial data set. When these algorithms are assessed in large size data sets, they suffer the scaling problem.

In the Section 3.1 the training set selection is presented. The Section 3.2 is dedicated to expose the scaling problem.

### 3.1. Training set selection

There may be situations in which there are too much data and these data in most cases are not equally useful in the training phase of the learning algorithm [25]. Instance selection mechanisms have been proposed to choose the most suitable points in the data set to become instances for the training data set used by the learning algorithm.

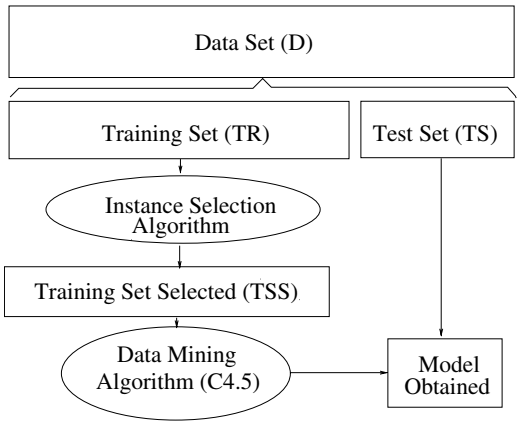


Fig. 1. Instance selection for training set selection.

In training set selection, the objective is to find training sets which can produce, when they are used as input, high precise and interpretable models.

The process, as we can see in Fig. 1 is the following: the initial data set ($D$) is divided in $TR$ and $TS$. Using $TR$ as input (learning set), the instance selection algorithms obtains the training set selected ($TSS$). The subset $TSS$ is used as input in the *C4.5* algorithm to generate its decision tree associated. This model will be validated using the test set $TS$.

In the following, we shortly revise the training sets selection approaches that we can find in the specialized literature. We classify them according to the model extracted after the training set selection process.

- Decision trees. In this group we can point:
  - In [24], Oates and Jensen study the effect of the training set size in decision trees complexity. The paper analyzes five decision tree pruning algorithms and the *Robust C4.5* algorithm as data reduction method. Authors reach as conclusions the relationship between tree size and training set size, where increasing training set size often results in a linear increase in tree size, even when that additional complexity does not improve the classification accuracy.
  - Sebban et al. in [5] apply training set selection to analyze the performances of the decision trees extracted from them. In this paper the interest is focused on the complexity and the generalization accuracy of the decision trees. Sebban et al. offer theoretical arguments to justify the data reduction techniques in favour of tree simplification, where some data reduction algorithms are very efficient to improve standard post-pruning performances.
- Neural networks. Training set selection has been used in the domain of neural networks:
  - In [25], a genetic algorithm is used for training data selection in radial based function networks. The approach is inspired in data editing concepts and outlier detection. Reeves and Bush apply a genetic algorithm to identify a 'good' training set for fitting radial basis function networks. They conclude that improved generalization can be obtained using this approach.
  - Valls et al. in [26] select training data to improve the generalization capabilities in radial basis neural networks. They propose a selective learning method in the domain of time-series prediction for a non-dimensional problem. In their approach, they consider that the amount of selected patterns or the neighbourhood choice around the new sample might influence in the generalization accuracy, and the neighbourhood must be established according to the dimensionality of the patterns.
- Different models. The training set selection is applied to extract quality subsets used as input to generate different sort of models:
  - Sierra et al. apply estimation of distribution algorithms, selecting instances and features for training set selection [27]. The subsets are evaluated by means of $k$-nearest neighbours, artificial neural networks and classification trees. The training set selection in this paper is applied to a medical problem. When the

resulting models are presented to the medical staff they noted that the confidence and acceptance of those models had increased.

- Cano et al. analyze evolutionary training set selection, comparing it with other non-evolutionary instance selection algorithms in [14]. The subsets extracted are evaluated as 1 nearest neighbour classifiers and by means of *C4.5* to generate decision trees. They combined the reduction rate and the 1 nearest neighbour precision of the subset selected in the fitness function to address the training set selection process. The conclusions reached indicate that evolutionary instance selection improves to non-evolutionary instance selection algorithms in the training set selection domain.
- Aguilar et al. in [28], and Riquelme et al. improving it in [29], apply training set selection based on ordered projection, analyzing the subsets using a $k$-nearest neighbour classifier and the *C4.5* algorithm. The study confirms that training set selection improves the efficiency of the models extractors and classifiers, and the accuracy and interpretability of the models and classifiers.
- In [30,31], Grochowski and Jankowski study different instance selection algorithms from the training set selection perspective. The first paper [30] presents a set of instance selection algorithms, which are evaluated as training set selectors in the second one [31]. The performance of the selected subsets is tested using $k$-nearest neighbours, support vector machine, *SSV* decision tree, a normalized version of *RBF* network called *NRBF*, *FSM* and *IncNet* model.
- Pedreira in [32] proposes a methodology to update Learning Vector Quantization prototypes by using a select subset of the available training data. The method selects, at each epoch, a subset of points considered to be at risk of being captured by another class prototype. The prototypes are updated only by the points that are under threat of being captured by a wrong prototype. A direct consequence of this procedure is that each prototype ends up located where it is really needed in order to defend its group feature vectors against the prototypes representing other groups. The results show some improvements if compared to the traditional *LVQ* update scheme.

### 3.2. The scaling problem

In this section we study the effect of the data set size in the instance selection algorithms and in the decision trees generated.

The majority of instance selection algorithms cannot deal with large size data sets. They have to face the following difficulties:

- Efficiency. The efficiency of non-evolutionary instance selection algorithms evaluated is at least of $O(n^2)$, being $n$ the number of instances in the data set. There are another set of algorithms (like *Rnn* in [33], *Snn* in [34], *Shrink* in [35], etc.) but most of them present an efficiency order much greater than $O(n^2)$. Logically, when the size grows, the time needed by each algorithm also increases.
- Resources. Most of the assessed algorithms need to have the complete data set stored in memory to carry out their execution. If the size of the data set was too big, the computer would need to use the disk as swap memory. This loss of resources has an adverse effect on efficiency due to the increased access to the disk.
- Generalization. Algorithms are affected in their generalization capabilities due to the noise and over fitting effect introduced by larger size data sets.
- Representation. *EAs* are also affected by representation, due to the size of their chromosomes. When the size of these chromosomes is so large, the algorithms experience converges difficulties, as well as costly computational time.

These drawbacks introduce considerable degradation in the behaviour of the instance selection algorithms. There is a group of them that cannot be evaluated due to its efficiency order (the case of *Snn* in [34] with $O(n^3)$).

On the other hand, algorithms evaluated directly on the whole larger data sets can be ineffective and/or inefficient.

## 4. Evolutionary stratified instance selection approach

The algorithm for the extraction of quality predictive models (high interpretable and precise) consists of the combination of the *EA* algorithm with the stratification of the initial data set to face the scaling problem. Following this way, the method could be applied to data sets independently of their size. The stratification reduces the search space, while the *EA* explores each strata.

The *EA* applied combines in its fitness function the accuracy offered by the 1-*Nearest Neighbour* classifier and the percentage of instances reduced. This situation makes us to consider the following:

- The use of this classifier to assess the classification percentage of the chromosomes introduces the scaling up problem in its evaluation, and the necessity of the stratification.
- The selection is developed by means of one classifier (1-*Nearest Neighbour*) which is not the one that is used to evaluate the classification performances of the final solution (*C4.5*).

As alternative, we introduce a new fitness function, where the classification performance of the chromosomes is assessed by means of *C4.5*, which is more efficient than 1-*Nearest Neighbour* so the stratification is needed just in very large size data sets. Using this fitness function, the selection is guided by the later classification algorithm (*C4.5*).

The Section 4.1 describes the use of *EAs* in training set selection, offering the solutions representation and both fitness functions considered. In Section 4.2, the evolutionary stratified instance selection applied in training set selection is presented.

### 4.1. Evolutionary algorithms applied in training set selection

The application of *EAs* to training set selection is accomplished by tackling two important issues: the specification of the representation of the solutions and the definition of the fitness functions.

#### 4.1.1. Representation

Let's assume a data set denoted *TR* with *n* instances. The search space associated with the instance selection is constituted by all the subsets of *TR*. Then, the chromosomes should represent subsets of *TR*. This is accomplished by using a binary representation. A chromosome consists of *n* genes (one for each instance in *TR*) with two possible states: 0 and 1. If the gene is 1, then its associated instance is included in the subset of *TR* represented by the chromosome. If it is 0, then this does not occur (see Fig. 2).

#### 4.1.2. Fitness function R-P

Let *TSS* be a subset (see Fig. 2) of instances of *TR* to evaluate and be coded by a chromosome. We define the fitness function that combines two values: the classification performance ($clasper_1$ with 1-Nearest Neighbour classifier) associated with *TSS* and the percentage of reduction ($percred_1$) of instances of *TSS* with regards to *TR* (this fitness function is denoted by *R-P*: Reduction-Precision):

$$Fitness_1(TSS) = \alpha \cdot clasper_1 + (1 - \alpha) \cdot percred_1. \quad (5)$$

The 1-*Nearest Neighbour* classifier is used for measuring the classification rate, $clasper_1$, associated with *TSS*. It denotes the percentage of correctly classified objects from *TR* using only *TSS* to find the nearest neighbour.

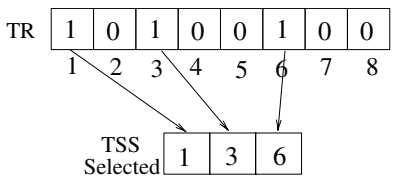


Fig. 2. Solutions representation.

For each object $y$ in $TR$, the nearest neighbour is searched for amongst those in the set $TSS \backslash \{y\}$. Whereas, $percred_1$ is defined as

$$percred_1 = 100 \cdot (|TR| - |TSS|)/|TR|. \tag{6}$$

The objective of the *EAs* is to maximize the fitness function defined, i.e., maximize the classification performance and minimize the number of instances obtained. In the experiments presented in this contribution, we have considered the value $\alpha = 0.5$ in the fitness function due to it presents the best trade off between reduction and accuracy in the final subsets selected.

#### 4.1.3. Fitness function I-P

Let $TSS$ be a subset (see Fig. 2) of instances of $TR$ to evaluate and be coded by a chromosome. The fitness function combines two values: the classification performance ($clasper_2$ with models extracted by *C4.5*) associated with $TSS$ and the percentage of reduction ($percred_2$) of decision tree size using as input $TSS$ with regards to $TR$ (this fitness function is denoted by *I-P*: Interpretability-Precision):

$$Fitness_2(TSS) = \alpha \cdot clasper_2 + (1 - \alpha) \cdot percred_2. \tag{7}$$

The models extracted by *C4.5* are used for measuring the classification rate, $clasper_2$, associated with $TSS$. It denotes the percentage of correctly classified objects from $TR$ by means of the decision tree generated using $TSS$ as input. Whereas, $percred_2$ is defined as

$$percred_2 = \frac{100 \cdot (SIZE_{TR} - SIZE_{TSS})}{SIZE_{TR}}. \tag{8}$$

The objective of the *EAs* is to maximize the fitness function defined, i.e., maximize the classification performance and minimize the size of the decision tree obtained. As in other fitness function, we have considered the value $\alpha = 0.5$ due to it presents the best trade off between decision tree size and accuracy in the final subsets selected.

### 4.2. Evolutionary stratified instance selection for training set selection

The stratified strategy has shown in previous works its behaviour facing the scaling problem [15]. It divides the initial data set in disjoint strata with equal class distribution. Due to the prototypes are independent one of each other, we can group them in these strata without loss of information.

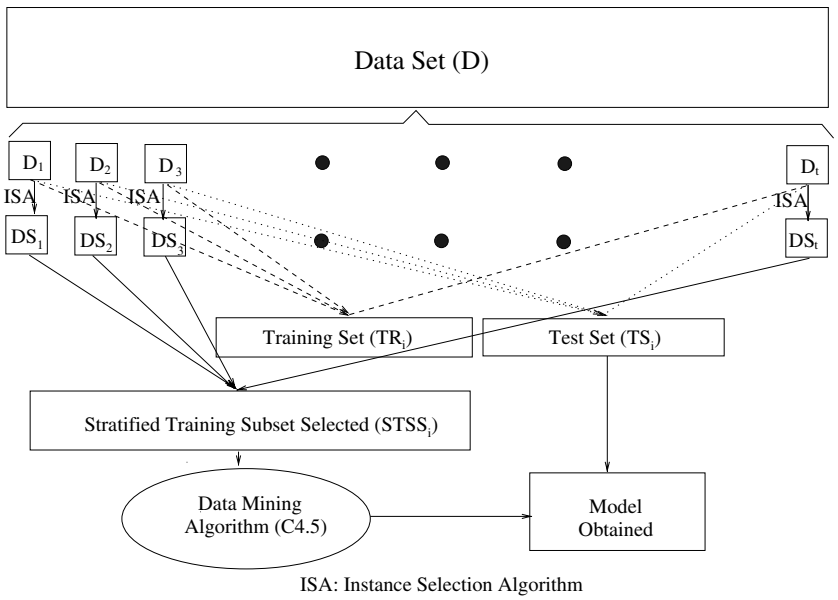


Fig. 3. Evolutionary stratified instance selection for training set selection.

The number of strata will determine the size of them. Using the proper number of strata we can reduce significantly the data set. This situation allows us to avoid the drawbacks suggested in Section 3.2.

Following the stratified strategy, initial data set $D$ is divided into $t$ disjoint sets $D_j$, strata of equal size, $D_1$, $D_2,\ldots,$ and $D_t$.

The test set $TS$ will be the $TR$ complementary one in $D$. The subsets $TR$ and $TS$ will be obtained as (9) and (10) show:

$$TR = \bigcup_{j\in J} D_j, \quad J \subset \{1,2,\ldots,t\}, \tag{9}$$

$$TS = D \setminus TR. \tag{10}$$

Instance selection algorithms (evolutionary and non-evolutionary) are applied in each $D_j$ obtaining a subset selected $DS_j$. The instance selected set ($TSS$) in stratified strategy is obtained using the $DS_j$ (see Eq. (11)) and it is called Stratified Training Subset Selected ($STSS$).

$$STSS = \bigcup_{j\in J} DS_j, \quad J \subset \{1,2,\ldots,t\}. \tag{11}$$

The complete process is presented in Fig. 3.

## 5. Experimental study

In this section we describe the experimental study developed. Section 5.1 shows the methodology followed in the experiments, Section 5.2 shows the results, finally, in the Section 5.3 we analyze them from different points of view (reduction, test accuracy, size of the model and balance precision-interpretability), using some statistical tests (ANOVA, Levene, Tamhane) for analyzing the algorithms accuracy.

### 5.1. Experimental methodology

In this subsection we present: the data sets, the algorithms assessed and their parameters, the stratification model.

#### 5.1.1. Data sets

We have carried out the experiments with increasing complexity and size of data sets. We have selected medium, large and huge size data sets as we can see in Tables 1–3 (these data sets can be found in the *UCI* Repository in [36], where the Kdd Cup'99 data set is particularly its 10% version).

Table 1
Medium size data sets

| Data set | Instances | Features | Classes |
|---|---|---|---|
| Pen-based recognition | 10,992 | 16 | 10 |
| SatImage | 6435 | 36 | 6 |
| Thyroid | 7200 | 21 | 3 |

Table 2
Large size data set

| Data set | Instances | Features | Classes |
|---|---|---|---|
| Adult | 30,132 | 14 | 2 |

Table 3
Very large size data set

| Data set | Instances | Features | Classes |
|---|---|---|---|
| Kdd Cup'99 | 494,022 | 41 | 23 |

*5.1.2. Algorithms and parameters*

The algorithms evaluated in this study will be divided in two groups, considering their evolutionary nature:

- Non-evolutionary algorithms. The algorithms selected will be: *Cnn* [37], *Ib2* [38], *Ib3* [38], which have been selected due to they are the most efficient non-evolutionary algorithms in [14], and John's Robust C4.5 [39], *PSRCG* [40] and *Random* which have shown the best behaviour in [5] to obtain quality training sets to extract the decision trees.
  The description of the algorithms is the following:
  - *Cnn* [37]: It tries to find a consistent subset, which correctly classifies all of the remaining points in the sample set. However, this algorithm will not find a minimal consistent subset.
  - *Ib2* [38]: It is similar to *Cnn* but using a different selection strategy.
  - *Ib3* [38]: It outperforms *Ib2* introducing the acceptable instance concept to carry out the selection.
  - *Robust C4.5* [39]: This algorithm removes interactively all instances misclassified by the current decision tree and builds a new one. It employs the *C4.5* algorithm to generate the decision trees.
  - *PSRCG* [40]: The algorithm considers a statistical information criterion based on a quadratic entropy computed from the nearest neighbour topology to carry out the remove of the instances.
  - *Random*: It selects randomly a training set fixing the reduction percentage it has to apply. This one has been added to compare the algorithms selection versus the random one.

  The parameters of *Ib3* are: Acceptance Level = 0.9 and Drop Level = 0.7. The other algorithms do not have parameters to be fixed.
- Evolutionary algorithms: We have selected the *CHC* [13] algorithm as efficient and effective model, due to its behaviour showed on [14]. The description of the algorithm is the following:
  During each generation the *CHC* algorithm 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. *HUX* exchanges half of the bits that differ between parents, where the bit position to be exchanged are randomly determined. *CHC* also employs a method of incest prevention. Before applying *HUX* on two parents, the Hamming distance between them is measured. Only those parents which 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. When no offspring are inserted into the new population the threshold is reduced by 1.
  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 members 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 re-seed the population. Re-seeding 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. Search is then resumed.
  The size of the population is 50 and the number of evaluations 10,000.

As reference we have introduced the *C4.5* algorithm using the initial data set without reduction, and following the ten fold cross-validation classic process (we denoted it *Tfcv cl*). When the size of the

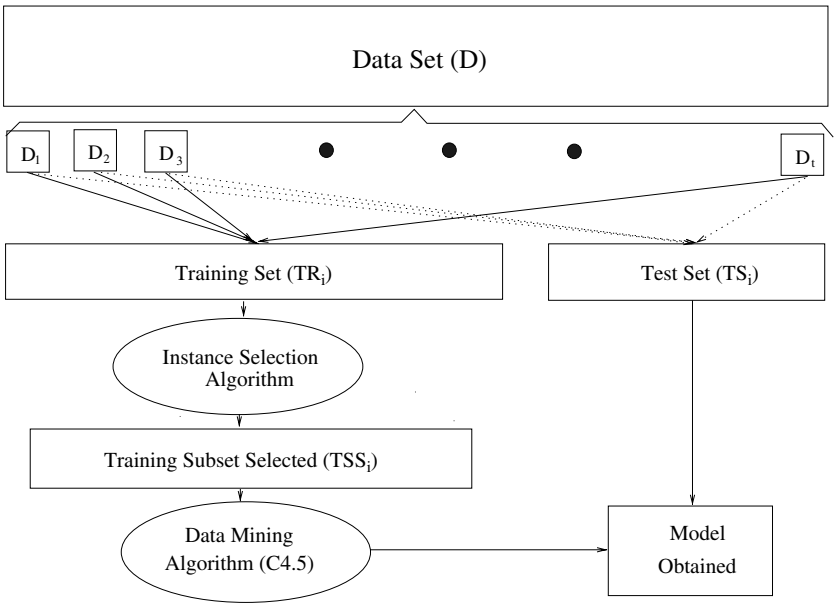


Fig. 4. Evolutionary instance selection for training set selection in Tfcv cl.

data sets permits us, we assess the instance selection algorithms over the complete data set in *Tfcv cl* (see Fig. 4).

We have included at the same time the execution of *C4.5* applying the maximal (*C4.5 Max*), minimal (*C4.5 Min*) and default (*C4.5*) Error-Based Prune to analyze the interpretability of the models generated.

As baseline, we have added to the experimentation the execution of the *Random* algorithm considering the minimal and the maximal reduction offered by the instance selection algorithms assessed.

### 5.1.3. Stratification and partitions

We have evaluated each algorithm in a ten fold cross-validation process. In the validation process $TR_i$, $|i| = 1,\ldots,10$ is a 90% of $D$ and $TS_i$ its complementary 10% of $D$.

The executions follow the model described in Fig. 3 called stratified Ten fold cross-validation (*Tfcv st*).

In *Tfcv st* each $TR_i$ and $TS_i$ are defined as we can see in (12) and (13), by means of the union of $D_j$ subsets.

$$TR_i = \bigcup_{j \in J} D_j, \quad J = \{j/1 \leqslant j \leqslant b \cdot (i-1) \quad \text{and} \quad (i \cdot b) + 1 \leqslant j \leqslant t\}, \tag{12}$$

$$TS_i = D \setminus TR_i, \tag{13}$$

where $t$ is the number of strata, and $b$ is the number of strata grouped ($b = t/10$, to carry out the ten fold cross-validation).

The $STSS_i$ subset is generated using the $DS_j$ instead of $D_j$ (see (14)).

$$STSS_i = \bigcup_{j \in J} DS_j, \quad J = \{j/1 \leqslant j \leqslant b \cdot (i-1) \quad \text{and} \quad (i \cdot b) + 1 \leqslant j \leqslant t\}. \tag{14}$$

$STSS_i$ contains the instances selected by instance selection algorithms in $TR_i$ following the stratified strategy.

For each data set we have employed the number of strata that appear in Table 4.

Table 4
Data sets stratification

| Pen-based recognition | SatImage | Thyroid | Adult | Kdd Cup'99 |
|---|---|---|---|---|
| $t = 10$ | $t = 10$ | $t = 10$ | $t = 100$ | $t = 100$ |

### 5.2. Results

In this section we describe and offer the tables where the results are shown.
The table presents the following structure:

- The first column shows the name of the algorithm. In this column the name is followed by the sort of validation process *st* (Tfcv st) or *cl* (Tfcv cl).
- The second column offers the average reduction percentage from the initial set.
- The third column contains the test accuracy associated to the decision tree classifier generated using the subset selected in stratification (*STSS*).
- The fourth column presents the number of rules which composed the model.
- The fifth column shows the mean number of antecedents of the rules of the model.
- The sixth column offers the time per algorithm execution consumed.

Tables 5–7 contain the results obtained in the evaluation of Pen-Based Recognition, SatImage and Thyroid data sets, respectively. In Table 8 we present the results obtained in the evaluation of Adult data set. Table 9 contains the results associated to Kdd Cup'99 data set.

### 5.3. Analysis

The analysis of Tables 5–9 is developed according to the following key points: Reduction percentage, Test Accuracy, Size of the model, balance Precision-Interpretability and Execution Time.

Table 5
Results associated to pen-based recognition data set

| | Red. | Test | Size | Ant. | Time |
|---|---|---|---|---|---|
| C4.5 Min cl | | 96.58 | 262.1 | 9.8 | 1 |
| C4.5 cl | | 96.46 | 185.2 | 9.5 | 1 |
| C4.5 Max cl | | 96.20 | 158.4 | 8.6 | 1 |
| Robust C4.5 cl | 3.54 | 96.46 | 176.6 | 9.2 | 11 |
| Robust C4.5 st | 3.31 | 96.20 | 168.4 | 8.9 | 2 |
| Cnn cl | 95.43 | 84.2 | 59.3 | 7.0 | 16 |
| $Random_1$ cl | 95.43 | 85.1 | 38.8 | 6.1 | 1 |
| Cnn st | 89.45 | 90.62 | 98.7 | 8.1 | 1 |
| Ib2 cl | 98.61 | 58.13 | 25.0 | 5.4 | 2 |
| $Random_2$ cl | 98.61 | 74.98 | 19.3 | 4.9 | 1 |
| Ib2 st | 94.31 | 79.30 | 48.5 | 6.2 | 1 |
| Ib3 cl | 96.39 | 80.9 | 57.6 | 6.8 | 7 |
| Ib3 st | 83.05 | 94.05 | 88.3 | 7.7 | 1 |
| PSRCG st | 94.95 | 75.97 | 42.1 | 6.3 | 63 |
| CHC I-P cl | 79.01 | 95.08 | 109.5 | 7.9 | 905 |
| $Random_3$ cl | 79.01 | 91.86 | 77.2 | 7.4 | 2 |
| CHC R-P st | 96.65 | 80.16 | 29.2 | 5.3 | 263 |
| $Random_4$ cl | 96.65 | 83.34 | 31.0 | 5.7 | 1 |

Table 6
Results associated to SatImage data set

| | Red. | Test | Size | Ant. | Time |
|---|---|---|---|---|---|
| C4.5 Min cl | | 86.27 | 444.2 | 12.4 | 1 |
| C4.5 cl | | 86.71 | 280.4 | 10.8 | 1 |
| C4.5 Max cl | | 87.59 | 144.3 | 9.0 | 1 |
| Robust C4.5 cl | 6.79 | 87.23 | 183.6 | 10.5 | 24 |
| Robust C4.5 st | 6.39 | 87.02 | 198.5 | 10.6 | 2 |
| Cnn cl | 80.26 | 78.36 | 183.7 | 12.5 | 30 |
| $Random_1$ cl | 80.26 | 82.98 | 83.0 | 8.3 | 1 |
| Cnn st | 75.12 | 80.44 | 208.6 | 11.7 | 1 |
| Ib2 cl | 96.50 | 52.18 | 32.2 | 7.8 | 3 |
| $Random_2$ cl | 96.50 | 77.04 | 19.8 | 5.3 | 1 |
| Ib2 st | 91.87 | 62.91 | 68.1 | 10.2 | 1 |
| Ib3 cl | 84.70 | 76.70 | 139.2 | 10.9 | 11 |
| Ib3 st | 78.11 | 86.49 | 186.7 | 10.7 | 1 |
| PSRCG st | 79.69 | 79.24 | 142.9 | 10.1 | 30 |
| CHC I-P cl | 51.2 | 84.98 | 115.3 | 9.3 | 25,240 |
| $Random_3$ cl | 51.2 | 84.1 | 135.5 | 8.9 | 2 |
| CHC R-P st | 94.32 | 78.83 | 15.5 | 4.4 | 128 |
| $Random_4$ cl | 94.32 | 79.03 | 30.0 | 6.3 | 1 |

Table 7
Results associated to Thyroid data set

| | Red. | Test | Size | Ant. | Time |
|---|---|---|---|---|---|
| C4.5 Min cl | | 99.01 | 38.4 | 7.6 | 1 |
| C4.5 cl | | 99.03 | 25.1 | 6.2 | 1 |
| C4.5 Max cl | | 99.06 | 10.8 | 4.2 | 1 |
| Robust C4.5 cl | 0.43 | 99.04 | 21.9 | 5.2 | 5 |
| Robust C4.5 st | 0.34 | 99.05 | 20.1 | 6.3 | 1 |
| Cnn cl | 81.25 | 97.32 | 13.1 | 4.7 | 31 |
| $Random_1$ cl | 81.25 | 98.63 | 10.6 | 4.7 | 1 |
| Cnn st | 78.93 | 98.78 | 14.0 | 4.9 | 1 |
| Ib2 cl | 99.22 | 93.71 | 3.2 | 1.6 | 1 |
| $Random_2$ cl | 99.22 | 94.19 | 2.0 | 1.5 | 1 |
| Ib2 st | 92.92 | 98.61 | 9.3 | 3.8 | 1 |
| Ib3 cl | 33.65 | 98.83 | 17.7 | 5.8 | 40 |
| Ib3 st | 38.62 | 99.01 | 22.2 | 7.0 | 1 |
| PSRCG st | 86.12 | 98.93 | 5.4 | 2.9 | 20 |
| CHC I-P cl | 50.44 | 99.16 | 7.9 | 3.4 | 7543 |
| $Random_3$ cl | 50.44 | 98.89 | 19.0 | 6.1 | 1 |
| CHC R-P st | 99.44 | 93.77 | 2.2 | 1.0 | 156 |
| $Random_4$ cl | 99.44 | 92.26 | 2.6 | 1.1 | 1 |

#### 5.3.1. Reduction percentage

Taking the second column of the tables into account, we can offer the following comments:

- The *Robust C4.5* is the algorithm which offers the smallest reduction over the initial data set. It cleans some noisy instances to improve the precision of the models extracted with *C4.5*.
- Among the non-evolutive instance selection algorithms, the one with the best behaviour in reduction is the *Ib2* in large and medium size data sets, followed by the *PSRGC* algorithm. In very large data sets as Kdd Cup'99, the best reduction is offered by *PSRGC*.
- In the *EAs* we can detect two different behaviours: *CHC I-P* is focused to the size of decision tree and its precision, so it is not interested in the number of instances and its reduction. For this reason, the reduction

Table 8
Results associated to Adult data set

| | Red. | Test | Size | Ant. | Time |
|---|---|---|---|---|---|
| C4.5 Min cl | | 84.02 | 1252.3 | 17.3 | 12 |
| C4.5 cl | | 85.4 | 359.8 | 14.3 | 11 |
| C4.5 Max cl | | 85.86 | 52.0 | 11.1 | 10 |
| Robust C4.5 cl | 12.16 | 85.69 | 297.1 | 11.9 | 37 |
| Robust C4.5 st | 11.52 | 86.15 | 193.3 | 12.8 | 1 |
| Cnn cl | 64.4 | 85.5 | 107.7 | 13.1 | 1 |
| $Random_1$ cl | 64.4 | 84.8 | 191.9 | 13.0 | 1 |
| Cnn st | 84.27 | 85.75 | 292.5 | 15.5 | 1 |
| Ib2 cl | 99.94 | 26.56 | 2.2 | 1.3 | 1 |
| $Random_2$ cl | 99.94 | 72.71 | 38.6 | 1.7 | 1 |
| Ib2 st | 99.57 | 36.4 | 12.1 | 5.0 | 1 |
| Ib3 cl | 79.42 | 83.76 | 145.9 | 12.2 | 3 |
| Ib3 st | 76.69 | 82.70 | 179.0 | 12.8 | 1 |
| PSRCG st | 96.84 | 75.77 | 47.8 | 8.1 | 4 |
| CHC I-P st | 58.54 | 85.24 | 203.5 | 13.5 | 108 |
| $Random_3$ cl | 58.54 | 85.09 | 216.3 | 14.5 | 1 |
| CHC R-P st | 99.38 | 82.7 | 5.9 | 2.8 | 38 |
| $Random_4$ cl | 99.38 | 79.89 | 13.9 | 5.5 | 1 |

Table 9
Results associate to Kdd Cup'99 data set

| | Red. | Test | Size | Ant. | Time |
|---|---|---|---|---|---|
| C4.5 Min cl | | 99.96 | 281.5 | 15.0 | 248 |
| C4.5 cl | | 99.95 | 143.8 | 11.7 | 375 |
| C4.5 Max cl | | 99.99 | 106.1 | 10.4 | 380 |
| Robust C4.5 st | 0.28 | 99.72 | 71.1 | 9.8 | 6 |
| Cnn st | 63.85 | 99.5 | 105.5 | 12.1 | 33 |
| $Random_1$ cl | 63.85 | 99.9 | 89.4 | 10.6 | 1 |
| Ib2 st | 82.01 | 95.05 | 58.2 | 10.8 | 21 |
| $Random_2$ cl | 82.01 | 99.9 | 61.9 | 9.3 | 1 |
| Ib3 st | 78.82 | 96.77 | 74.3 | 11.4 | 2 |
| PSRCG st | 99.88 | 98.6 | 37.0 | 7.6 | 5634 |
| CHC I-P st | 60.32 | 99.7 | 69.3 | 10.0 | 1306 |
| $Random_3$ cl | 60.32 | 99.9 | 94.7 | 10.8 | 1 |
| CHC R-P st | 99.28 | 98.41 | 9.5 | 3.5 | 4912 |
| $Random_4$ cl | 99.28 | 99.44 | 18.8 | 6.1 | 1 |

rate of *CHC I-P* is average. On the other hand, *CHC R-P* combines the reduction rate as objective in the fitness function, so it presents high reduction rates.The *CHC R-P* shows high reduction rates in the data sets analyzed, independently of the size of them. It presents reduction rates greater than 94% in all cases.

### 5.3.2. Test accuracy

To compare the results provided by *C4.5* over the different training set selection algorithm outputs we develop a statistical analysis. First, we use the ANOVA analysis of one factor [16] for each problem to be used for that purpose; the factor being the algorithms used. Given that significant differences were found for all algorithms with respect to the mean result values associated with the different algorithms analyzed, we performed a Tamhane means rank test [16] with a confidence coefficient of 95%, as the hypothesis of equality of variances of the results was rejected in all of the analyzes performed for each method (Levene test). The tests were performed using *SPSS* [41] statistical package (see from Tables A.1–A.3 in Appendix A).

Table 10
Resume of Tamhane test having CHC I-P as reference focused on precision

| Algorithm | Better or equal in precision |
|---|---|
| C4.5 Min cl | 3/5 |
| C4.5 cl | 3/5 |
| C4.5 Max cl | 3/5 |
| Robust C4.5 cl | 3/4 |
| Robust C4.5 st | 2/5 |
| Cnn cl | 4/4 |
| $Random_1$ cl | 4/5 |
| Cnn st | 4/5 |
| Ib2 cl | 4/4 |
| $Random_2$ cl | 4/5 |
| Ib2 st | 5/5 |
| Ib3 cl | 4/4 |
| Ib3 st | 4/5 |
| PSRCG st | 5/5 |
| $Random_3$ cl | 4/5 |
| CHC R-P st | 5/5 |
| $Random_4$ cl | 5/5 |

Table 10 resumes the tables offered in Appendix A where the Tamhane test for each problem, having *CHC I-P* as reference, is applied. The first column is the name of the algorithm which is being compared to *CHC I-P*. The second one represents the number of data sets where the algorithm *CHC I-P* presents a better or equal behaviour than the algorithm which is in the first column.

Considering the third column in Tables 5–10, we can point out that:

- The *CHC I-P* presents one of the highest test precision rates among the instance selection algorithms studied, near to the *C4.5* ones. According to the statistical analysis, we can point out that the *CHC I-P* precision is better or equal than the offered by most of instance selection algorithms. We have an exception with *Robust C4.5* that produces small data reduction.
- In the *EAs* case, the *CHC I-P* offers better precision rates than *CHC R-P* due to the first one has associated smaller reduction rates to generate the models.

*5.3.3. Size of the model*

The size of the model can be studied considering the fourth and fifth columns of the results tables (Tables 5–9), corresponding to the mean number of rules and the mean number of antecedents per rule. We can point out the following:

- Usually, the size of the predictive models is related to the size of the input training data set used to generate them. The instance selection algorithms which present the best reduction rates are often the ones that present the smaller predictive models. We have an exception with the *PSRCG* algorithm, which presents high reduction rates with medium size models.
- The biggest decision trees correspond to the *C4.5* executions, with maximal, minimal or default prune, and Robust *C4.5*.
- Among the non-evolutionary instance selection algorithms, the best one is *Ib2*, which has associated high reduction rate, but it presents very bad test accuracy.
- Focusing our attention on the *EAs*, *CHC R-P* offers the minimal decision trees due to its maximal reduction percentage. The average reduction rate in *CHC I-P* produces that its model associated is bigger than the one generated by *CHC R-P*.The *CHC R-P* generates one of the minimal decision trees when the size of data set grows. In the fourth and fifth columns of Table 9, dedicated to the biggest data set (Kdd Cup'99), we can see that *C4.5* with maximal prune obtains models with 106.1 rules and 10.4 antecedents while stratified *CHC R-P* reduces the size to 9.5 rules and 3.5 antecedents per rule.

Table 11
Resume of Tamhane test having CHC I-P as reference, considering precision and size

| Algorithm | Better or equal in precision | Better or equal in size of the model (smaller size) |
|---|---|---|
| C4.5 Min cl | 3/5 | 5/5 |
| C4.5 cl | 3/5 | 5/5 |
| C4.5 Max cl | 3/5 | 4/5 |
| Robust C4.5 cl | 3/4 | 4/4 |
| Robust C4.5 st | 2/5 | 4/5 |
| Cnn cl | 4/4 | 2/4 |
| $Random_1$ cl | 4/5 | 2/5 |
| Cnn st | 4/5 | 4/5 |
| Ib2 cl | 4/4 | 0/4 |
| $Random_2$ cl | 4/5 | 0/5 |
| Ib2 st | 5/5 | 1/5 |
| Ib3 cl | 4/4 | 2/4 |
| Ib3 st | 4/5 | 3/5 |
| PSRCG st | 5/5 | 1/5 |
| $Random_3$ cl | 4/5 | 4/5 |
| CHC R-P st | 5/5 | 0/5 |
| $Random_4$ cl | 5/5 | 0/5 |

- Comparing the decision trees extracted from the *Random* selection, we can point out that *CHC R-P* improves considerably the results of the *Random* selection, with smaller models in all the data sets assessed.

Due to the size of the model affects directly to the interpretability of the model, we can consider that *CHC R-P* offers the most interpretable decision trees.

*5.3.4. Balance precision-interpretability*

In this study the objective considered is the analysis of the extraction of highly precise-interpretable prediction models by means of instance selection algorithms. Having the precision and the interpretability (size of the models) key points in mind we add Table 11. This table presents the behaviour relationship in accuracy test and interpretability between *CHC I-P* and the rest of algorithms.

The conclusions reached analyzing Tables 5–9 and 11 are the following:

- The models generated without reduction by means of *C4.5* have the highest accuracy rates, but their decision trees are the biggest ones, so their interpretability is reduced.
- The models associated to the *Random* selection present high accuracy rates, but their models are bigger than the ones extracted from the *CHC R-P* selection.
- The best behaviour in interpretability belongs to *Ib2*, due to its high reduction rate, but the precision it has associated is very poor.
- The *CHC R-P* and *CHC I-P* present a high trade off between precision and interpretability. The *CHC R-P* produces smaller models than *CHC I-P*, but the last one offers higher accuracy rate, near to the associated to *C4.5* without reduction.

*5.3.5. Execution time*

Paying attention to the execution time of the algorithms we can offer the following comments (Tables 5–9):

- As the size of data set grows, the *C4.5* execution time grows too. The proper reduction of the data set improves the execution time of *C4.5*. To increase the prune rate of *C4.5* affects to the execution time negatively.
- The non-evolutionary algorithms present smaller computational cost than *CHC* due to the evolutionary process that *CHC* has associated.
- Between both *CHC* versions, the execution of *CHC R-P* is the fastest one.

The execution time associated to *CHC* represents a greater cost than the offered by the non-evolutionary algorithms, however its application is interesting because it produces a high trade off between test accuracy and small size of the decision trees generated.

## 6. Concluding remarks

In this contribution we have analyzed the extraction of classification rule-based models by means of evolutionary stratified training set selection. The quality of the models has been evaluated considering their accuracy and interpretability.

The main conclusions reached are the following:

- The evolutionary stratified instance selection (*CHC R-P*) offers the best model size, maintaining an acceptable accuracy. It produces the smallest set of rules, with the minimal number of rules and the smallest number of antecedents per rule.
- The stratified *CHC I-P* allows us to obtain models with high test accuracy rates, similar to *C4.5*, but with the advantage of the size of the models that are reduced considerably.

Finally, we can conclude that the predictive model extraction by means of evolutionary stratified training set selection (*CHC R-P* or *I-P*) presents a good trade off between accuracy and interpretability. Our proposals present a very good scaling up behaviour, obtaining good results when the size of data set grows.

## Acknowledgements

The authors are very grateful to our colleague and friend Cesar Hervas who have offered us his invaluable knowledge to develop the statistical analysis.

## Appendix A

See Tables A.1–A.3.

Table A.1
Averaged values, standard deviations, mean differences and critical values of the Tamhane test of the results of Pen-Based and SatImage data sets for CHC I-P

| Algorithm | Pen based | | | | SatImage | | | |
|---|---|---|---|---|---|---|---|---|
| | Mean | SD | Mean diff. | *p*-Value | Mean | SD | Mean diff. | *p*-Value |
| C4.5 Min cl | 96.581 | 0.20648 | 1.5010(*) | 0 | 86.27 | 0.16773 | −1.2893(*) | 0 |
| C4.5 cl | 96.46 | 0.16138 | 1.3800(*) | 0 | 86.71 | 0.12463 | −1.7293(*) | 0 |
| C4.5 Max cl | 96.2 | 0.19385 | 1.1200(*) | 0 | 87.59 | 0.11343 | −2.6093(*) | 0 |
| Robust C4.5 cl | 96.462 | 0.3255 | 1.3820(*) | 0 | 87.231 | 0.38304 | −2.2503(*) | 0 |
| Robust C4.5 st | 96.2 | 0.54371 | 1.1200(*) | 0.006 | 87.02 | 0.41085 | −2.0393(*) | 0 |
| Cnn cl | 84.201 | 0.74912 | 10.8790(*) | 0 | 78.359 | 0.58463 | 6.6217(*) | 0 |
| $Random_1$ cl | 85.141 | 1.01902 | 9.9390(*) | 0 | 82.98 | 1.34582 | 2.0007(*) | 0 |
| Cnn st | 90.619 | 0.58911 | 4.4610(*) | 0 | 80.44 | 0.87932 | 4.5407(*) | 0 |
| Ib2 cl | 58.13 | 1.48531 | 36.9500(*) | 0 | 52.181 | 0.99849 | 32.7997(*) | 0 |
| $Random_2$ cl | 74.9797 | 1.2403 | 20.1003(*) | 0 | 77.0387 | 1.50062 | 7.9420(*) | 0 |
| Ib2 st | 79.299 | 0.8464 | 15.7810(*) | 0 | 62.908 | 1.38984 | 22.0727(*) | 0 |
| Ib3 cl | 80.9 | 0.51584 | 14.1800(*) | 0 | 76.7 | 0.78771 | 8.2807(*) | 0 |
| Ib3 st | 94.049 | 0.42459 | 1.0310(*) | 0.001 | 86.49 | 0.68532 | −1.5093(*) | 0.005 |
| PSRCG st | 75.969 | 0.59207 | 19.1110(*) | 0 | 79.239 | 0.48303 | 5.7417(*) | 0 |
| $Random_3$ cl | 91.861 | 1.14942 | 3.2190(*) | 0 | 84.1007 | 1.02602 | 0.8800(*) | 0.017 |
| CHC R-P st | 80.159 | 1.30844 | 14.9210(*) | 0 | 78.83 | 0.34143 | 6.1507(*) | 0 |
| $Random_4$ cl | 83.3403 | 1.28502 | 11.7397(*) | 0 | 79 | 1.20198 | 5.9807(*) | 0 |

(*) In Tamhane test, if $p < 0.05$, then the mean difference associated presents a significative value.

Table A.2
Averaged values, standard deviations, mean differences and critical values of the Tamhane test of the results of Thyroid and Adult data sets for CHC I-P

| Algorithm | Thyroid | | | | Adult | | | |
|---|---|---|---|---|---|---|---|---|
| | Mean | SD | Mean diff. | $p$-Value | Mean | SD | Mean diff. | $p$-Value |
| C4.5 Min cl | 99.01 | 0.01491 | 0.149 | 0.502 | 84.021 | 0.19902 | 1.2180(*) | 0 |
| C4.5 cl | 99.03 | 0.01414 | 0.129 | 0.854 | 85.401 | 0.16135 | −0.162 | 1 |
| C4.5 Max cl | 99.061 | 0.00876 | 0.098 | 1 | 85.859 | 0.08306 | −0.6200(*) | 0.001 |
| Robust C4.5 cl | 99.04 | 0.02667 | 0.119 | 0.963 | 85.69 | 0.32338 | −0.451 | 0.334 |
| Robust C4.5 st | 99.05 | 0.03801 | 0.109 | 0.996 | 86.151 | 0.23082 | −0.9120(*) | 0 |
| Cnn cl | 97.319 | 0.31409 | 1.8400(*) | 0 | 85.499 | 0.3276 | −0.26 | 1 |
| $Random_1$ cl | 98.629 | 0.17103 | 0.5300(*) | 0 | 84.8007 | 0.1692 | 0.4383(*) | 0.028 |
| Cnn st | 98.781 | 0.18953 | 0.3780(*) | 0.01 | 85.751 | 0.20179 | −0.5120(*) | 0.012 |
| Ib2 cl | 93.71 | 1.40509 | 5.4490(*) | 0 | 26.56 | 0.46205 | 58.6790(*) | 0 |
| $Random_2$ cl | 94.1893 | 1.51943 | 4.9697(*) | 0 | 72.7103 | 1.62002 | 12.5287(*) | 0 |
| Ib2 st | 98.61 | 0.25586 | 0.5490(*) | 0.004 | 36.402 | 2.58145 | 48.8370(*) | 0 |
| Ib3 cl | 98.829 | 0.28661 | 0.33 | 0.604 | 83.76 | 0.65501 | 1.4790(*) | 0.004 |
| Ib3 st | 99.009 | 0.07894 | 0.15 | 0.744 | 82.701 | 0.58966 | 2.5380(*) | 0 |
| PSRCG st | 98.931 | 0.18114 | 0.228 | 0.59 | 75.77 | 0.93051 | 9.4690(*) | 0 |
| $Random_3$ cl | 98.89 | 0.22215 | 0.2690(*) | 0.012 | 85.09 | 0.25365 | 0.149 | 1 |
| CHC R-P st | 93.77 | 0.33407 | 5.3890(*) | 0 | 82.7 | 0.37818 | 2.5390(*) | 0 |
| $Random_4$ cl | 92.2597 | 1.04137 | 6.8993(*) | 0 | 79.8903 | 0.7984 | 5.3487(*) | 0 |

(*) In Tamhane test, if $p < 0.05$, then the mean difference associated presents a significative value.

Table A.3
Averaged values, standard deviations, mean differences and critical values of the Tamhane test of the results of Kdd Cup'99 data set for CHC I-P

| Algorithm | Kdd Cup'99 | | | |
|---|---|---|---|---|
| | Mean | SD | Mean diff. | $p$-Value |
| C4.5 Min cl | 99.962 | 0.01229 | −0.2610(*) | 0 |
| C4.5 cl | 99.95 | 0.01247 | −0.2490(*) | 0 |
| C4.5 Max cl | 99.99 | 0 | −0.2890(*) | 0 |
| Robust C4.5 st | 99.72 | 0.05578 | −0.019 | 1 |
| Cnn st | 99.5 | 0.06912 | 0.2010(*) | 0.001 |
| $Random_1$ cl | 99.899 | 0.02734 | −0.1980(*) | 0 |
| Ib2 st | 95.049 | 0.14761 | 4.6520(*) | 0 |
| $Random_2$ cl | 99.8993 | 0.0297 | −0.1983(*) | 0 |
| Ib3 st | 96.769 | 0.27674 | 2.9320(*) | 0 |
| PSRCG st | 98.6 | 0.28802 | 1.1010(*) | 0 |
| $Random_3$ cl | 99.8997 | 0.02371 | −0.1987(*) | 0 |
| CHC R-P st | 98.41 | 0.18547 | 1.2910(*) | 0 |
| $Random_4$ cl | 99.4403 | 0.16816 | 0.2607(*) | 0 |

(*) In Tamhane test, if $p < 0.05$, then the mean difference associated presents a significative value.

## References

[1] I.H. Witten, E. Frank, Data mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, 2000.
[2] M. Last, O. Maimon, A compact and accurate model for classification, IEEE Transactions on Knowledge and Data Engineering 16 (2) (2004) 203–215.
[3] Kweku-Muata, Osei-Bryson, Evaluation of decision trees: a multicriteria approach, Computers and Operations Research 31 (2004) 1933–1945.
[4] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
[5] M. Sebban, R. Nock, J.H. Chauchat, R. Rakotomalala, Impact of learning set quality and size on decision tree performances, International Journal of Computers, Systems and Signals 1 (1) (2000) 85–105.
[6] H. Liu, H. Motoda, On issues of instance selection, Data Mining and Knowledge Discovery 6 (2002) 115–130.

[7] D.R. Wilson, T.R. Martinez, Reduction techniques for instance-based learning algorithms, Machine Learning 38 (2000) 257–268.
[8] T. Back, Evolutionary Algorithms in Theory and Practice, Oxford University Press, 1996.
[9] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.
[10] D.E. Goldberg, The Design of Competent Genetic Algorithms: Steps Toward a Computational Theory of Innovation, Kluwer Academic Publishers, 2002.
[11] L. Kuncheva, Editing for the *k*-nearest neighbors rule by a genetic algorithm, Pattern Recognition Letters 16 (1995) 809–814.
[12] H. Shinn-Ying, L. Chia-Cheng, L. Soundy, Design of an optimal nearest neighbour classifier using an intelligent genetic algorithm, Pattern Recognition Letters 23 (13) (2002) 1495–1503.
[13] L.J. Eshelman, The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination, Foundations of Genetic Algorithms 1 (1991) 265–283.
[14] 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.
[15] J.R. Cano, F. Herrera, M. Lozano, Stratification for scaling up evolutionary prototype selection, Pattern Recognition Letters 26 (7) (2005) 953–963.
[16] T.W. Anderson, An Introduction to Multivariate Statistical Analysis, John Wiley and Sons, 1984.
[17] M. Bohanec, I. Bratko, Trading accuracy for simplicity in decision trees, Machine Learning 15 (1994) 223–250.
[18] L. Hall, R. Collins, K. Bowyer, R. Banfield, Error-based pruning of decision trees grown on very large data sets can work! in: International Conference on Tools for Artificial Intelligence, 2002, pp. 233–238.
[19] T. Oates, D. Jensen, Large data sets lead to overly complex models: an explanation and a solution, in: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, 1998, pp. 294–298.
[20] C. Schaffer, When does overfitting decrease prediction accuracy in induced decision trees and rule sets? in: Proceedings of the European Working Session on Learning (EWSL-91), 1991, pp. 192–205.
[21] Z.-H. Zhou, Y. Jiang, Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble, IEEE Transactions on Information Technology in Biomedicine 7 (1) (2003) 37–42.
[22] L.A. Breslow, D.W. Aha, Simplifying decision trees: a survey, Knowledge Engineering Review 12 (1) (1997) 1–40.
[23] F. Esposito, D. Malerba, G. Semeraro, A comparative analysis of methods for pruning decision trees, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (5) (1997) 476–491.
[24] T. Oates, D. Jensen, The effects of training set size on decision tree complexity, in: Proceedings of the Fourteenth International Conference on Machine Learning, 1997, pp. 254–262.
[25] C.R. Reeves, D.R. Bush, Using genetic algorithms for training data selection in RBF networks, in: Instance Selection and Construction for Data Mining, Kluwer Academic Publishers, 2001, pp. 339–356.
[26] J.M. Valls, I.M. Galvan, P. Isasi, How the selection of training patterns can improve the generalization capability in radial basis neural networks, Applied Informatics (2003) 275–280.
[27] B. Sierra, E. Lazkano, I. Inza, M. Merino, P. Larranaga, J. Quiroga, Prototype selection and feature subset selection by estimation of distribution algorithms. A case study in the survival of cirrhotic patients treated with tips, in: Proceedings of the Eighth Conference on AI in Medicine, Lecture Notes in Computer Science, Springer, 2001, pp. 20–30.
[28] J.S. Aguilar, J.C. Riquelme, M. Toro, Data set editing by ordered projection, Intelligent Data Analysis 5 (5) (2001) 405–417.
[29] J. Riquelme, J. Aguilar, M. Toro, Finding representative patterns with ordered projections, Pattern Recognition 36 (4) (2003) 1009–1018.
[30] M. Grochowski, N. Jankowski, Comparison of instance selection algorithms I. Algorithms survey, in: Proceedings of the Seventh International Conference on Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, vol. 3070, Springer, 2004, pp. 580–585.
[31] M. Grochowski, N. Jankowski, Comparison of instance selection algorithms II. Results and comments, in: Proceedings of the Seventh International Conference on Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, vol. 3070, Springer, 2004, pp. 598–603.
[32] C.E. Pedreira, Learning vector quantization with training data selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (1) (2006) 157–161.
[33] G.W. Gates, The reduced nearest neighbour rule, IEEE Transaction on Information Theory 18 (5) (1972) 431–433.
[34] G.L. Ritter, H.B. Woodruff, S.R. Lowry, T.L. Isenhour, An algorithm for a selective nearest neighbour decision rule, IEEE Transaction on Information Theory 21 (6) (1975) 665–669.
[35] D. Kibbler, D.W. Aha, Learning representative exemplars of concepts: an initial case of study, in: Proceedings of the Fourth International Workshop on Machine Learning, 1987, pp. 24–30.
[36] C.B.D.J. Newman, S. Hettich, C. Merz, UCI repository of machine learning databases, 1998. Available from: <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
[37] P.E. Hart, The condensed nearest neighbour rule, IEEE Transaction on Information Theory 18 (3) (1968) 431–433.
[38] D. Aha, D. Kibbler, M. Albert, Instance-based learning algorithms, Machine Learning 6 (1991) 37–66.
[39] G.H. John, Robust decision trees: removing outliers from databases, in: Proceedings of the First Conference on Knowledge Discovery and Data Mining, 1995, pp. 174–179.
[40] M. Sebban, R. Nock, Instance pruning as an information preserving problem, in: Proceedings of the Seventeenth International Conference on Machine Learning, 2000, pp. 855–862.
[41] I. SPSS, SPSS 11.0 Advanced Models, SPSS Inc., 1999.

**José Ramón Cano** received the M.Sc. and Ph.D. degrees in computer science from the University of Granada, Granada, Spain, in 1999 and 2004, respectively. He is currently an Associate Professor in the Department of Computer Science, University of Jaén, Jaén, Spain. His research interests include data mining, data reduction, interpretability-accuracy trade off, and evolutionary algorithms.

**Francisco Herrera** received the M.Sc. degree in Mathematics in 1988 and the Ph.D. degree in Mathematics in 1991, both from the University of Granada, Spain. He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has published over 100 papers in international journals and he is coauthor of the book "*Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*" (World Scientific, 2001). As edited activities, he has co-edited three international books and co-edited 15 special issues in international journals on different Soft Computing topics, such as, "*Preference Modelling*", "*Computing with Words*", "*Genetic Algorithms*" and "*Genetic Fuzzy Systems*". He currently serves on the editorial boards of the Journals: Soft Computing, Mathware and Soft Computing, International Journal of Hybrid Intelligent Systems, and International Journal of Computational Intelligence Research. His current research interests include computing with words, preference modelling, data mining and knowledge discovery, data reduction, fuzzy rule-based systems, genetic algorithms, and genetic fuzzy systems.

**Manuel Lozano** received the M.Sc. and Ph.D. degrees in computer science from the University of Granada, Granada, Spain, in 1992 and 1996, respectively. He is currently an Associate Professor with the Department of Computer Science and Artificial Intelligence, University of Granada. His research interests include, machine learning, genetic algorithms and data reduction.