Evolutionary Stratified Instance Selection applied to Training Set Selection for Extracting High Precise-Interpretable Classification Rules [*]

José Ramón Cano
University of Jaén
Dept. of Computing Science
E.P.S. Linares, 23700, Linares, Jaén, Spain
jrcano@ujaen.es

Francisco Herrera and Manuel Lozano
University of Granada
Dept. of Comp. Science and Artificial Intelligence
E.T.S. Ingen. Informática, 18071, Granada, Spain
(herrera , lozano) @decsai.ugr.es

Abstract

The generation of predictive models is a frequent task in data mining with the objective of generating high 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 contribution, we analyze the predictive model extraction based on rules using a training selected set via evolutionary stratified instance selection. This method face to the scaling up problem that appears in the evaluation of large size data sets.

1. Introduction

A basic process in data mining is the generation of representative models from data [22]. The models, depending of their domain of application, can be:

- Predictive models. The objective of these models is the accuracy or precision of the model. In the literature we can find different proposals to measure the quality of the predictive models, like simplicity, interpretability, etc [13].

- Descriptive models. This sort of models try to find relationships and behavior patterns in the data set which offer knowledge in the *DM* problem.

In this contribution we are going to focus our attention in the predictive models based on classification rules for large size data sets under a preprocessing process via data reduction. The models have been extracted from the data sets by means of *C4.5* algorithm [17]. We can use different measures to analyze the quality of the tree generated [13].

A possible way to improve the behavior of predictive models (precision and interpretability) is to extract them from suitable reduced/selected training sets. Training set selection can be developed using instance selection algorithms. The instance selection algorithms select representative instance subsets following a determined selection strategy, and they can improve the nearest neighbor rule prediction capabilities [14, 21].

Evolutionary algorithms (*EAs*) are adaptable methods based on natural evolution that can be applied in search and optimization problems [2, 7, 8]. The *EAs* offer interesting results when they are assessed on instance selection [12, 20]. In this study, we use *CHC* algorithm [5] as *EA* considering its behavior shown in [4].

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

We focus our attention in 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. The stratification reduces the original data set size, splitting it in 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.

The outline of the contribution is the following. In Section 2 we analyze the predictive models and their extraction using *C4.5*, presenting the measures considered to assess their behavior. Section 3 describes the drawbacks that the evaluation of very large data sets introduced in instance selection algorithms and in decision tree extraction. Section 4 presents the evolutionary stratified instance selection process applied to training set selection. Section 5 contains the experimental study developed, offering the methodol-

[*] This work was supported by project TIC2002-04036-C05-01.

ogy followed, the results and their analysis. Finally, in Section 6 we reach the conclusions.

2. Predictive Models: Classification trees extraction with C4.5

The importance of decision trees and rules is that they are favored techniques for building 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. Specifically, each branch of the tree is a classification question and the leaves of the tree are partitions where they present the data set with their classification.

In this study we are going to extract the decision trees using the *C4.5* algorithm [17].

The models generated are complete and consistent, covering all the examples of the training set. This situation produces that the models over fit the training set and reduce their precision when they classify new instances. In addition, the models are sensible to the presence of noise in the training sets, adjusting their branch and nodes to it. To limit these drawbacks, it is applied prune methods to the decision trees generated [16]. The 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.

The prune methods increase the generalization capabilities of the model, and reduce its size, which increase its interpretability.

The drawback for both prune methods, preprune and postprune, is to determine the stop limit. The limit will depend of the training set where the decision tree is being extracted. The proper adjust of the limit produces model with better or worse behavior. 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.

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. The measures we are going to use to assess the predictive models extracted with *C4.5* will be the following [13].

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)):

$$N_{Antec} = \sharp|Cond| \tag{3}$$

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

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

3. Scaling Problem

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

The majority of instance selection algorithms cannot deal with large size data sets. In this section we study the effect of the data set size in the instance selection algorithms.

The main difficulties they have to face are the following:

- 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 [6], *Snn* in [18], *Shrink* in [11], 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 algorithms assessed 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 convergence difficulties, as well as costly computational time.

These drawbacks introduce considerable degradation in the behaviour of instance selection algorithms. There is a group of them that can't be evaluated due to its efficiency order (the case of *Snn* in [18] with $O(n^3)$).

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

The size of decision trees generated using large size data sets as input is increased considerably [3, 9, 23]. The high size of the decision tree produces:

- Over fitting. In this case, the learned hypothesis is so closely related to the training examples such its generalization capabilities would be penalized [19].

- Low Human interpretability. The high size of the decision tree introduces the disadvantage of excessive complexity that can render it incomprehensible to experts [13, 24].

4. Evolutionary Stratified Instance Selection Approach

To carry out the instance selection we have combined the stratification of the initial data set with *EAs*. Following this way, the method could be applied to data sets independently of their size. The stratification reduces the search space, while the *EAs* explore each strata.

The Subsection 4.1 shows the training set selection process. In the Subsection 4.2 we describe the use of *EAs* in training set selection, offering the solutions representation and the fitness function. Finally, in Subsection 4.3, the evolutionary stratified instance selection applied in training set selection is presented.

4.1. Training Set Selection

The objective is to find training sets which can produce, when they are used as input, high precision and interpretable set of rules (see Figure 1).

The initial data set (*D*) is divided in *TR* and *TS*. Using *TR* as input, 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 validate using the set *TS*.

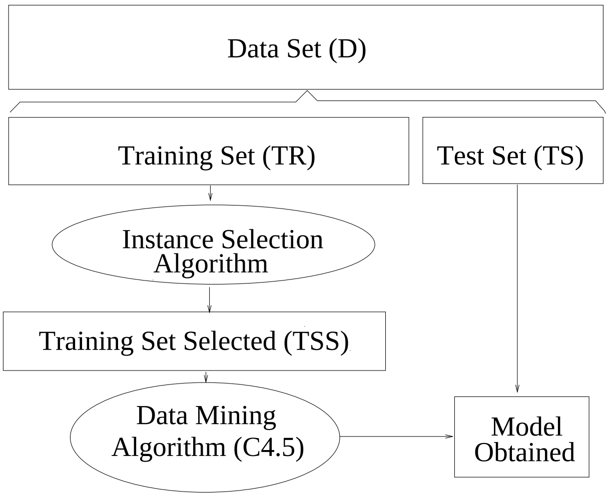

Data Set (D)
Training Set (TR)
Test Set (TS)
Instance Selection Algorithm
Training Set Selected (TSS)
Data Mining Algorithm (C4.5)
Model Obtained

Figure 1. Prototype Selection for Training Set Selection

4.2. Evolutionary Algorithms applied in Training Set Selection

The application of *EAs* to instance selection is accomplished by tackling two important issues: the specification of the representation of the solutions and the definition of the fitness function [4].

4.2.1 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.

4.2.2 Fitness Function

Let *TSS* be a subset (see Figure 1) of instances of *TR* to evaluate and be coded by a chromosome. We define the fitness function that combines two values: the classification performance ($clasper$) associated with *TSS* and the percentage of reduction ($percred$) of instances of *TSS* with regards to *TR*:

$$Fitness(TSS) = \alpha \cdot clasper + (1 - \alpha) \cdot percred. \quad (5)$$

The 1-*NN* classifier is used for measuring the classification rate, $clasper$, associated with *TSS*. It denotes the percentage of correctly classified objects from *TR* using only

TSS to find the nearest neighbor. For each object *y* in *TR*, the nearest neighbor is searched for amongst those in the set $TSS \setminus \{y\}$. Whereas, $percred$ is defined as:

$$percred = 100 \cdot (|TR| - |TSS|)/|TR|. \qquad (6)$$

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 balance between reduction and accuracy in the final subsets selected.

4.3. Evolutionary Stratified Instance Selection for Training Set Selection

The stratified strategy 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.

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.

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

The test set *TS* will be the *TR* complementary one in *D*. The subsets *TR* and *TS* will be obtained as (7) and (8) show:

$$TR = \bigcup_{j \in J} D_j, J \subset \{1, 2, ..., t\} \qquad (7)$$

$$TS = D \setminus TR \qquad (8)$$

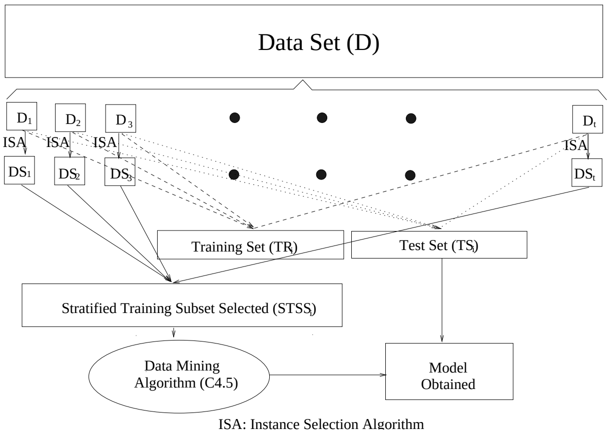
Instance selection algorithms (evolutionary and non-evolutionary) are applied in each D_j obtaining a subset selected DS_j. The instance selected set (*TSS*) in stratified strategy is obtained using the DS_j (see equation (9)) and it is called Stratified Training Subset Selected (*STSS*).

$$STSS = \bigcup_{j \in J} DS_j, J \subset \{1, 2, ..., t\} \qquad (9)$$

The complete process is presented in Figure 2:

5 Experimental Study

In this section we describe the experimental study developed. Subsection 5.1 shows the methodology followed in the experiments, Subsection 5.2 shows the results, finally, in the Subsection 5.3 we analyze them.



Data Set (D)
D1
D2
D3
Dt
ISA
ISA
ISA
ISA
DS1
DS2
DS3
DSt
Training Set (TR)
Test Set (TS)
Stratified Training Subset Selected (STSS)
Data Mining Algorithm (C4.5)
Model Obtained
ISA: Instance Selection Algorithm

Figure 2. Evolutionary Stratified Instance Selection for Training Set Selection

5.1 Experimental Methodology

In this subsection we present the data set, the algorithms studied with their parameters, the stratification model and the partitions, and the C4.5 experimentation.

5.1.1 Data Set, Algorithms and Parameters

The data set used in the experiments is the *KDD Cup*'99, concretely its 10% version. It presents large size so we can use it to analyze the scaling problem. This data set contains 494022 instances, with 41 attributes and 23 different classes (this data set can be found in the *UCI* Repository in [15].

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* [10], *Ib2* [1], *Ib3* [1]. They have been selected due to they are the non evolutionary algorithms most efficient ones in [4]. The parameters of *Ib3* are: Aceptance Level=0,9 and Drop Level=0,7. The other algorithms don't have parameters to be fixed.

- Evolutionary Algorithms: We have selected the *CHC* algorithm as efficient and efficacy model, due to its behavior showed on [4]. The size of the population is 50 and the number of evaluations 10000.

5.1.2 Stratification and Partitions

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

The executions follow the model described in Figure 2. We call it stratified Ten fold cross validation (*Tfcv st*).

In *Tfcv st* each TR_i and TS_i are defined as we can see in (11) and (11), by means of the union of D_j subsets.

$$TR_i = \bigcup_{j \in J} D_j, J = \{j/1 \leq j \leq b \cdot (i-1) \quad (10)$$
$$and\ (i \cdot b) + 1 \leq j \leq t\}$$

$$TS_i = D \setminus TR_i \quad (11)$$

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 (13)).

$$STSS_i = \bigcup_{j \in J} DS_j, J = \{j/1 \leq j \leq b \cdot (i-1) \quad (12)$$
$$and\ (i \cdot b) + 1 \leq j \leq t\}$$

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

The number of strata used is *t*=100 for *KDD Cup*'99.

5.1.3 On the C4.5 experimentation

As reference we introduce 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*).

We have included at the same time the execution of *C4.5* applying the maximal (*C4.5 Max*) and minimal (*C4.5 Min*) prune to analyze the interpretability of the models generated.

5.2 Results

In this section we describe and offer the table 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* and the number of strata for *Tfcv st*, or *cl* meaning ten fold cross validation classic process (*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 Table 1 contains the results for *KDD Cup*'99 data set.

Table 1. Results for Kdd Cup'99.

	RED	TEST	SIZE	ANT
C4.5 Min cl		99,96	252	13,34
C4.5 cl		99,95	143	11,78
C4.5 Max cl		99,95	102	10,52
Cnn st	81,61	96,43	83	11,49
Ib2 st	82,01	95,05	58	10,86
Ib3 st	78,82	96,77	74	11,48
CHC st	99,28	98,41	9	3,56

5.3 Analysis

The analysis of Table 1 allow us to make the following comments:

- Considering the reduction percentage, the evolutionary stratified instance selection presents the best behavior among the algorithms studied.
- The evaluation of *C4.5* in the original data set without reduction offers the best test accuracy percentages. The stratified *CHC* shows the best results among the instance selection algorithms.
- The size of predictive model can be related to the size of the input training data set used to generate it.

  The instance selection algorithms which present the best reduction rates are often the ones that present the smaller predictive models. The stratified *CHC* offers the minimal training set selected and their models associated are the smallest. In the fourth column of Table 1 we can see that *C4.5* with maximal prune obtains models with 102 rules and 10,52 antecedents while stratified *CHC* reduces the size to 9 rules and 3,52 antecedents per rule.

  The size of the model affects directly to the interpretability of the model.

6 Conclusions

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

The principal conclusion reached are the following:

- The evolutionary stratified instance selection offers the best reduction percentages from the initial data set.
- The stratified *CHC* shows the best test accuracy rates among the instance selection algorithms studied.
- Paying attention to the size of the model, the stratified *CHC* produces the smaller set of rules, with the minimal number of rules and the smallest number of antecedents per rule. The evolutionary stratified instance selection offers the most interpretable predictive models.

As concluding remark, we consider that the predictive model extraction by means of evolutionary stratified training set selection presents the best behavior among the instance selection algorithms studied. It offers the minimal predictive models with higher accuracy rates, similar than the associated to *C4.5* without reduction. The stratified *CHC* permits to obtain the predictive models with the best balance between interpretability and precision.

References

[1] D. Aha, D. Kibbler, and M. Albert. Instance-based learning algorithms. *Machine Learning*, 6, 1991.

[2] T. Back, D. Fogel, and Z. Michalewicz. *Handbook of evolutionary computation*. Oxford University Press, 1997.

[3] M. Bohanec and I. Bratko. Trading accuracy for simplicity in decision trees. *Machine Learning*, 15:223–250, 1994.

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

[5] L. J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. *Foundations of Genetic Algorithms*, 1:265–283, 1991.

[6] G. W. Gates. The reduced nearest neighbour rule. *IEEE Transaction on Information Theory*, 18(5):431–433, 1972.

[7] D. Goldberg. *The design of competent genetic algorithms: Steps toward a computational theory of innovation*. Kluwer Academic Pub., 2002.

[8] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.

[9] L. Hall, R. Collins, K. Bowyer, and R. Banfield. Error-based pruning of decision trees grown on very large data sets can work! In *International Conference on Tools for Artificial Intelligence*, pages 233–238, 2002.

[10] P. E. Hart. The condensed nearest neighbour rule. *IEEE Transaction on Information Theory*, 18(3):431–433, 1968.

[11] D. Kibbler and D. W. Aha. Learning representative exemplars of concepts: An initial case of study. In *Proc. of the Fourth International Workshop on Machine Learning*, pages 24–30, 1987.

[12] L. Kuncheva. Editing for the k-nearest neighbors rule by a genetic algorithm. *Pattern Recognition Letters*, 16:809–814, 1995.

[13] Kweku-Muata and Osei-Bryson. Evaluation of decision trees: a multicriteria aproach. *Cumputers and Operations Research*, 31:1933–1945, 2004.

[14] H. Liu and H. Motoda. On issues of instance selection. *Data Mining and Knowledge Discovery*, 6:115–130, 2002.

[15] C. J. Merz and P. M. Murphy. UCI repository of machine learning databases. 1996. University of California Irvine, Department of Information and Computer Science, http://kdd.ics.uci.edu.

[16] J. Mingers. An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, 4(2):227–243, 1989.

[17] J. R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, 1993.

[18] G. L. Ritter, H. B. Woodruff, S. R. Lowry, and T. L. Isenhour. An algorithm for a selective nearest neighbour decision rule. *IEEE Transaction on Information Theory*, 21(6):665–669, 1975.

[19] 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)*, pages 192–205, 1991.

[20] H. Shinn-Ying, L. Chia-Cheng, and L. Soundy. Design of an optimal nearest neighbour classifier using an intelligent genetic algorithm. *Pattern Recognition Letters*, 23(13):1495–1503, 2002.

[21] D. R. Wilson and T. R. Martinez. Reduction tecniques for instance-based learning algorithms. *Machine Learning*, 38:257–268, 2000.

[22] I. H. Witten and E. Frank. *Data mining: practical machine learning tools and techniques with java implementations*. Morgan Kaufmann, 2000.

[23] Z. Zheng. Scaling up the rule generation of C4.5. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 348–359, 1998.

[24] Z.-H. Zhou and Y. Jiang. Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble. *IEEE Transactions on Information Technology in Biomedicine*, 7(1):37–42, 2003.