AI Communications 18 (2005) 63–65
IOS Press

Thesis

Improving the performance of evolutionary algorithms for decision rule learning *

Raúl Giráldez Rojo
Department of Computer Science, University of Seville, Spain
E-mail: giraldez@lsi.us.es

Abstract. Evolutionary algorithms appear as an interesting alternative to achieve minimal error rates and low numbers of rules in supervised learning tasks. In spite of the computational cost of this approach, some proposals can be applied to make the algorithm faster and more efficient. This paper describes some of these proposals, which are integrated in the evolutionary tool HIDER*. Specifically, we developed a new genetic encoding for the individuals of the evolutionary population and a novel data structure for the evaluation process. These approaches allow the evolutionary algorithms to reduce the high computational cost and to obtain high quality solutions.

Keywords: Supervised learning, evolutionary algorithms, decision rules

1. Introduction

Supervised Learning is used when we want to build a knowledge model from a training labelled dataset and predict the outcome of a new unseen instance. In this field, the knowledge models are usually represented as decision rules or trees. A decision rule is a "*if* $\mathcal{C}$ *then* $\mathcal{L}$" type, where $\mathcal{C}$ is a conjunction of conditions that establishes what values the attributes can take to classify an example with class label $\mathcal{L}$.

Evolutionary Computation techniques are commonly used to address problems with very large search space, where an exhaustive search is not applicable in practice [10]. Thus, machine learning tasks, such as decision rule discovery, have been solved by using Evolutionary Algorithms (henceforth EAs) [2,5,12]. In this context, the selection of a suitable representation for the individuals (encoding) and a appropriate fitness function (evaluation) are two critical factors in applications based on EAs.

The aim of our research was to improve the performance of EAs for decision rule discovery in two directions: *efficacy*, by biasing the search towards better solutions, and *efficiency*, by reducing the computational cost of the algorithms. In order to achieve such goals, we focused our research on the aforementioned critical factors. With respect to the encoding, we developed a new approach called *Natural Coding* [7], including its own genetic operators. On the other hand, the data structure *EES* (Efficient Evaluation Structure) [8] was designed specifically for accelerating the evaluation process of individuals during the running of the EA. All these approaches are integrated in the evolutionary tool named HIDER*, whose main characteristics are summarized in next sections.

*The research was supported by the Spanish Research Agency CICYT under grant TIN2004-00159.

2. HIDER*: Hierarchical Decision Rules

The main algorithm is a typical sequential covering method, where the algorithmic function that produces the rules is an EA. The individuals of this EA are potential decision rules encoded by natural coding. The EA returns one rule every time it is called and such rule is used to eliminate examples from the training data. Thus, HIDER* generates rules sequentially until all the examples of the training data are covered.

2.1. Natural coding

Natural coding [7] handles efficiently continuous and discrete attributes by encoding each attribute with

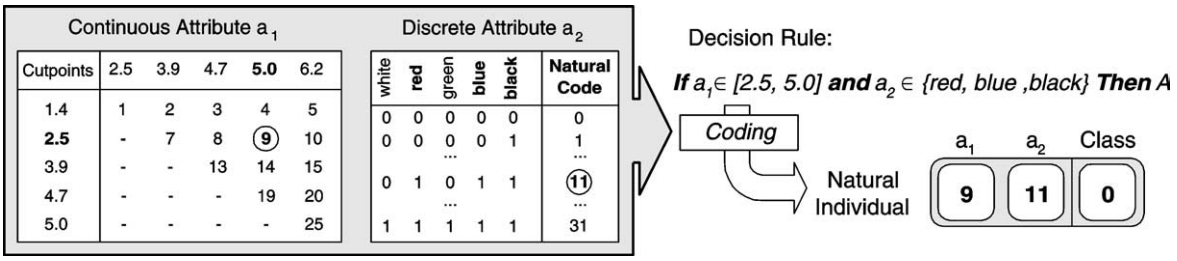

Fig. 1. Natural coding.

only one gene, that is, a natural number. This coding guarantees the two following properties: uniqueness (every individual has an unique representation) and minimality (the length of coding must be as short as possible).

For continuous attributes, a discretization method is applied in order to decrease the cardinality of the set of continuous values, theoretically infinite. Specifically, we used an original discretizer, named USD [6], which generates a set of cutpoints that maximizes the global accuracy of the intervals and whose quality is proved in [3]. Once the cutpoints are calculated, a natural number is assigned to any possible combination of bounds. In case the attributes are discrete, the natural coding is obtained from a binary coding similar to that used in others EAs [2] by transforming the binary string into decimal representation. Finally, for the class, the natural coding is simply an enumeration of the labels. Figure 1 shows an example of natural coding for a decision rule with one continuous attribute and other discrete. With respect to the genetic operators, the mutation and crossover are algebraic operations with negligible computational cost. For example, the natural mutation of the kth bit of a gene n for discrete attributes is shown in Eq. (1),

$$mut_k(n) = \left(n + 2^{k-1}\right)\%2^k + 2^k \left\lfloor \frac{n}{2^k} \right\rfloor \qquad (1)$$

where $k \in 1, 2, \ldots, |A|$; $|A|$ is the number of values of the attribute; $\%$ is the rest of the integer division; and $\lfloor\_\rfloor$ is the integer part.

Natural coding allows the reduction of the search space size, makes the genetic operators more accurate, and also allows to decrease the number of generations and population size. These improvements were experimentally proven by means of comparisons with hybrid coding (joining binary and real codings) [2]. Thus, natural coding allows the EA to decrease the error rate and the number of rules of the knowledge model using only one third of the generations and fewer than three fourth of the population size used by the version with hybrid coding.

2.2. EES: Efficient Evaluation Structure

Learning methods usually evaluate the rules directly from the database. That is to explore such database sequentially, taking each of the examples and testing the quality of the rule through the correct classification of those examples. We can deduce, therefore, that the evaluation process of these systems is very costly in terms of time and space. This aspect is even more significant when the learning method is an EA, mainly due to the repetitive evaluation of the candidate solutions.

We considered how to design a data structure that incorporates knowledge over the distribution of the examples in the attribute hyper-space [9]. This information is very useful to locate the regions that must be explored in such space, and evaluate the individuals only in these regions, and thus to speed up the evaluation process. In this sense, EES [8] organises the information from a database in such a way that it allows us to process only those examples whose values are covered by such rule, instead of the whole database.

EES is a vector of binary and balanced search-trees where the ith element of the vector contains information about the ith attribute (a_i) in the database. Depending on the type of the attribute a_i, each node in the ith tree stores a value (discrete) or interval (continuous) and a list with the indexes of those examples whose ith attribute corresponds to the value of the node. In this way, the evaluation process of examples is incremental, i.e., it starts from a number of covered examples that is reduced as we analyse more attributes of the structure. Thus EES calculates what examples are within a rule, if any; and does not check weather each example satisfies or not all the conditions of the rule.

3. Empirical results

In order to check the performance of our approach, any number of tests were carried out on several databases from the UCI Repository [4] and the results of a 10-fold cross-validation were compared to those obtained by the previous version [2] and others non-evolutionary classifiers as C4.5 and C4.5Rules [11].

HIDER* presented better performance than the others classification tools. It reduced significantly the number of rules of most of databases used in the experiments without damaging the classification accuracy. On average, our tool generated a number of rules six times smaller than C4.5 and half than C45Rules. With respect to the previous version with hybrid coding, HIDER* also obtained rules with better quality using only 23% of the computational resources. Furthermore, in all of the experiments, the structure EES provided improvements on the times obtained using the linear method, resulting in a computational reduction of about 52.4%.

4. Conclusions

The high computational cost associated with rule learning systems based on EAs is reduced by means of a new way to encode the individuals of the genetic population and a novel data structure which speeds up the evaluation process. Natural coding allows the reduction of the search space size, making the search for decision rules faster. In addition, the genetic operators are provided, as simple algebraic expressions. On the other hand, the structure EES organises the information in such a way that only the necessary examples from the database will be dealt with, and not all data.

Acknowledgements

The author is grateful to José C. Riquelme and Jesús S. Aguilar-Ruiz for supervising his doctoral dissertation.

References

[1] J.S. Aguilar-Ruiz, Removing examples and discovering hierarchical decision rules with evolutionary algorithms, *Artificial Intelligence Communications* **14**(4) (2003), 231–233.

[2] J.S. Aguilar-Ruiz, J.C. Riquelme and M. Toro, Evolutionary learning of hierarchical decision rules, *IEEE Transactions on Systems, Man and Cybernetics – Part B* **33**(2) (2003), 324–331.

[3] J.S. Aguilar-Ruiz, J. Bacardit and F. Divina, Experimental evaluation of discretization schemes for rule induction, in: *Genetic and Evolutionary Computation Conference – GECCO 2004*, Part I, D. Kalyanmoy et al., eds, LNCS 3102, Springer, Seattle, Washington, USA, 2004, pp. 828–839.

[4] C. Blake and E.K. Merz, *UCI Repository of Machine Learning Databases*, 1998.

[5] F. Divina and E. Marchiori, Evolutionary concept learning, in: *Genetic and Evolutionary Computation Conference – GECCO 2002*, W.B. Langdon et al., eds, Morgan Kaufmann, NY, USA, 2002, pp. 343–350.

[6] R. Giráldez, J.S. Aguilar-Ruiz and J.C. Riquelme, Discretization oriented to decision rule generation, in: *Knowledge-Based Intelligent Information Engineering Systems and Applied Technologies*, Part I, E. Damiani et al., eds, IOS Press, Crema, Italy, 2002, pp. 275–279.

[7] R. Giráldez, J.S. Aguilar-Ruiz and J.C. Riquelme, Natural coding: a more efficient representation for evolutionary learning, in: *Genetic and Evolutionary Computation Conference – GECCO 2003*, Part II, E. Cantú-Paz et al., eds, LNCS 2724, Springer, Chicago, USA, 2003, pp. 279–290.

[8] R. Giráldez, J.S. Aguilar-Ruiz, J.C. Riquelme and D. Mateos, An efficient data structure for decision rules discovery, in: *ACM Symposium on Applied Computing SAC'03 – Track on Data Mining*, G. Lamont et al., eds, ACM, Melbourne, FL, USA, 2003, pp. 475–479.

[9] R. Giráldez, J.S. Aguilar-Ruiz and J.C. Riquelme, Knowledge-based fast evaluation for evolutionary learning, *IEEE Transactions on Systems, Man and Cybernetics – Part C*, 2005 (in press).

[10] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[11] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA, 1993.

[12] G. Venturini, SIA: a supervised inductive algorithm with genetic search for learning attributes based concepts, in: *European Conference on Machine Learning*, P. Brazdil, ed., LNCS 667, Springer, Vienna, Austria, 1993, pp. 281–296.