# ALTERNATIVE METHOD OF CLASSIFICATION BASED ON TREES DECISION

**MARÍA DE GUADALUPE COTA**
University of Sonora
lcota@gauss.mat.uson.mx

**JUAN PABLO SOTO**
University of Sonora
jpsoto@gauss.mat.uson.mx

## ABSTRACT

*The objective of the inductive learning is to build a model to discover new patterns based on training set designed by experts of the item concerned, formed with characteristics and data of problems solved in the past and similar to those raised. In this work describes an algorithm which presents original contributions based on the use of pattern recognition techniques supervised in the stage of pre-processing of information, as well as in the construction of decision trees and the generation and evaluation of rules. The 2G algorithm produces new knowledge that they represent through rules with a different structure from that obtained by traditional trees decision-making, since the algorithm uses a minimum of attributes and only requires two conditions to evaluate new instances. Such contributions are related to the concept of "virtual class", "breaking-up of classes" and " virtual rules no explicit". The results obtained improve to the reported in examples of literature and 2G is considered a stable algorithm since it does not show considerable variation when applied to different problems.*

*Key words: classification, pattern recognition, supervisor*

## 1. INTRODUCTION

The decision trees are an alternative method that can be used as predictive models for decision making in artificial intelligence applications. Are constructed using a knowledge base structured by experts and logical constructs are generated to represent the conditions that constitute a set of rules that can be applied in solving complex problems.

This paper presents 2G algorithm based on decision trees [1] [16], which has proved efficient in treating problems with continuous attributes. The main differences compared with other algorithms of the same type is that generates trees with shallow and procedures to generate rules with no more than two conditions by time of evaluation. Moreover, in the generation of rules applies the concept of "virtual class" to include attributes values that have ambiguity in existing classes. In the procedure of classification creates the concept of "no explicit virtual patterns" to classify instances where not all attributes involved qualify in the same rule, and can be classified with different rules for each one of the attributes, with the only restriction that the selected rules should belong to the same class

To evaluate the performance of the algorithm were selected some classic problems and results of the tests were compared with results of open source algorithms such as C4.5 [1] [2], See5 [3] and J48 [4]. Additionally we compared the results with those obtained with methodologies listed above that are posted on official Web sites [5] [6] [7] [8] [9] [10] [11] [12]. It is important to note that 2G produces better results than those published in the above references.

For organizational purposes, this article is divided into four sections: introduction, description of the algorithm 2G, the results of tests, work to future and conclusions.

## 2. 2 ALGORITHM DESCRIPTION

The concept of 'pattern' is the main element in the area of pattern recognition, which is identified as a set of properties associated with an object type. A pattern may classify a set of predefined values or can be inferred in a sample space based on similarities found [13].

Today, automated techniques are considered very important to extract knowledge from data banks that cannot be analyzed manually by human beings because they are large, but mostly because the obtained data can be used as support in decision making, as they reflect situations that have happened in the past and present, and form the basis for the application of pattern recognition techniques in the process of discovering new knowledge [14].

Such concepts are applied in the classification procedure, where objects are grouped based on similarity criteria in categories for finding a set of models (or functions) that describe the classes of instances for which is unknown to which class belongs [13].

In this context the decision trees are tools used to represent knowledge through supervised recognition, and are considered highly efficient methods in decision-making under uncertainty. The method applied uses inductive learning, building a classifier to discover new patterns extracted from the training set which is formed by a set of attributes to produce a model for classifying new examples on the basis of rules that are generated based on path branches of the tree built, starting from the root node until reaching the leaves (classes) [15] [18].

Given this, the algorithm 2G is an alternative method for constructing decision trees on a specific issue, which selects the most important attributes of the training set by applying the concept of entropy and the variability of the data, the creation of rules that can be reduced according to a procedure we call "class break pattern", applying a treatment to change the next class in the case of continuous values that are representative bodies for which the class is not known to which they belong, a problem that occurs when the same value which may belong to different classes.

In the 2G algorithm, the attribute selection is done by applying the concept of entropy to obtain the information gain [21], [2]. To determine this, it is necessary to have a set of values for each attribute, which are grouped by a measure that identify each group of values with same class. In this case are chosen the continuous values, the maximum of each group and it is applied the process with base in the following ideas and concepts:

- Domain.- Name that allows identify a set of values with homogeneous properties [17].
- Discourse universe (DU).- Environment where it is defined the problem represented by the Cartesian product of a set of finite domains [17].
- Attribute or variable of a problem.- Object with its own characteristics that can take values existing in the DU related to a certain domain [17].
- Training example.- Represented by a row of the attributes set.
- Class.- Defines the generalization of a particular object and represents a pattern or a prototype of a family of concrete objects. An instance of the class is represented trough a training example [17].

The possible distribution of attribute values can coincide with one of the following patterns (see figure 1):

1. The only attribute values belong to a class. 2. The attribute values overlap more than one distinct class.

3. The values are outside the scope of classes, but within the universe of values included. The case of points 2 and 3 are those that generate problems in the identification of classes, since it prevents a precise classification of new instances as they are values that lie in the range inaccurate

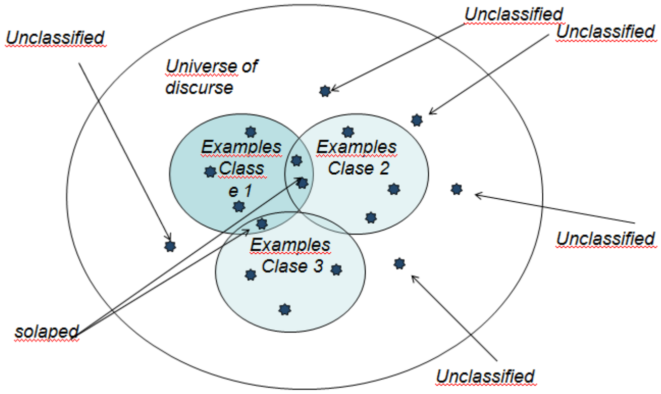


Fig. Distribution of values

A set of patterns is considered as a group if it has homogeneous and has different characteristics of the rest in the sample space of representation. To accomplish this are applied the specification describe below:

*S* original set of training examples
*Sr* reduced set of training examples
*p* total number of *S* or *Sr* examples
*e* instance of *p*
*c* class to which belongs *e*
*k* total number of classes
*z* index of the class value *c* related to *e*

and *z* = {1, ...., *c*}}
*n* total number of examples of a class *e*i
*ll* number of attributes of the set *S* or *Sr*
*A* set of attributes
*Ar* selected set of attributes from *G(Ar)*
*t total number of examples of A or Ar*
*v* set of possible values that can take *A* in the discourse of the universe or from *Ar*
*m* number of examples of *v*
*T* number of values of *v* in *A* or *Ar*
*probi* probability that an example (*ei*) of *Ar* belongs to a class *c* with i = {1, ...., t}

The entropy set H(*Sr*), the entropy of attributes H(*Ar*), the entropy of values H(v) and the attributes gain of the reduced set G(*Ar*), which are applied through the following expressions:

$$H(Sr) = \sum_{i=1}^{c} - prob_i * log_2 (prob_i)$$

$$H(Ar) = \sum^{T} H(Sr) - prob_i *$$

$$H(v) = \sum^{t} - prob_i * log_2$$

$$G(Ar) = \sum_{x=1}^{t} H(S) - H(Ar)$$

The model applies the classic methodology described in the literature [21] [2]. Receive as input a set of training examples that are processed to build a decision tree with elimination of well classified examples and generate rules to validate new instances of the problem in question

To discretize the values of attributes, we are based on methods used in algorithms like C4.5 [1] [2],, See5 [3] and J48 [4], but for best results we use distinct operations, which are described below:

One of the traditional methods of selection of representative values of an attribute is to select a set of values that are at the point where a change is detected in the vector class, then choose the average values of the adjacent values that meet these criteria, ignoring repeated values and implement a process to reduce the resulting list, to which is applied a heuristic selection or statistical estimate as the mean, median, etc. An example of this is shown in Table I, which describes the values for the attribute 'A'. The values involved in sampling and averaging the values found in items where a class change is detected are marked with an 'x', obtaining the following vector: (67.5, 85.0, 95.0).

As an example with the traditional method it would be ordered a vector of values of the Table I:

TABLE I.- EXAMPLE OF DISCRETIZATION OF VALUES. TRADITIONAL METHOD

| V | 65 | 70 | 70 | 80 | 80 | 90 | 99 | 100 |
|---|---|---|---|---|---|---|---|---|
| C | -1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 |
| S | x |  |  |  |  | x |  | x |

V = value of attribute 'A'
C = class
S = value selected

In our case, for greater accuracy in the generation of rules, about us select for each attribute the values that are the point where it changes of class vector to what we call “breaking of the class” an then eliminate redundancies (Table II). Later, it is selected the maximum value for each group of values continuous of the same class to form a set of representative values per attribute. As an exception, and unlike traditional methods, we include the maximum value of the groups of values that present “overlapping classes”, identifying them as groups of values that belong to “virtual classes” allowing them to participate in the selection of values and being part of the set of values that will eventually be used in the attributes selection stage. To make the final selection of representative values it was constructed a matrix where the values of the resultant vector of representative values selected by every attribute are set as column headings. For each class is added a row, and in the internal cells of the matrix where the column value matches with the class value it is used the variable “ninst” that can take the value of 1 if there exists at least one example that meets the properties of the row and column, and the value 0 if otherwise.

An example of this is shown in the section highlighted in gray in Table II. So that the resultant vector for us is: (65, 80, 99, 100), which has a positive impact when applying the algorithm 2G because important values rejected by other methods are now included.

TABLE II.- EXAMPLE OF DISCRETIZATION OF VALUES. ALGORITHM 2G.

| Value | 65 | 70 | 80 | 85 | 86 | 90 | 99 | 100 |
|---|---|---|---|---|---|---|---|---|
| Class -1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| Class 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
|  | x |  | x |  | x |  | x | x |

The procedure implemented in 2G algorithm is described to following:

a) Information recorded:

- Classes (*c*)
- Attributes (*Ar*)
- Attribute values (*v*)
- Training examples (*e*)

b) Calculation of values using (1):

- *p*, *k*, *n*, *l*, *t*, *m*, *T*.
- Combined entropy H(*Sr*).
- H(*Ar*) it is applied for each attribute
- G(*Ar*) it is applied just one time and the four attributes that present the biggest degree of information gain are selected.

c) Building of the decision tree (*Ar*):

The biggest values representatives of the attributes are taken for each combination of class. The level of the tree is limited to four and the permutation between the first and second selected node is realized.

d) Rules generation

The rules are generated only with two conditions and for each rule are fixed the minimum and maximum values.

e) Evaluation of the test set.

To evaluate new examples it is taken as reference the tree structure.

During the process of assignment of class there are examples for which do not exist rules that can classify them. To treat this type of instances, these are allowed to be classified with more than one rule, only if they belong to the same class. If even so, some examples cannot conclude this procedure it is possible to classify these examples using the attribute that provides greater information gain and assign to them the class of the rule that matches a very close value to the maximum value of the minimum recorded for the rule.

Below is a brief description on the application of the procedure called "no explicit virtual patterns" (see figure 3):

I) We selected the first class of the attribute that has a higher gain.

II) An instance can be classified assigning different rules to its attributes, but must be of the same class. An example of this is shown in Figure 1, where C+ and C- are the classes and sections in gray are the different rules. The arrows indicate the route of evaluation of new instances that are classified with this technique.

Sample 1 is classified by a rule of class C+ and Sample 2 is classified with two rules of class C+. Moreover, Rn represents the number of rule with n = 1, 2..., N. When we cannot classify the values of an instance in one class, we choose the attribute that has more gain of information and analyze three options:

a. Select the class that has the first rule.

b. Choose the rule that has the minimum value of the minimum value (see figure 2).

c. Select the majority class of the rule set candidates.

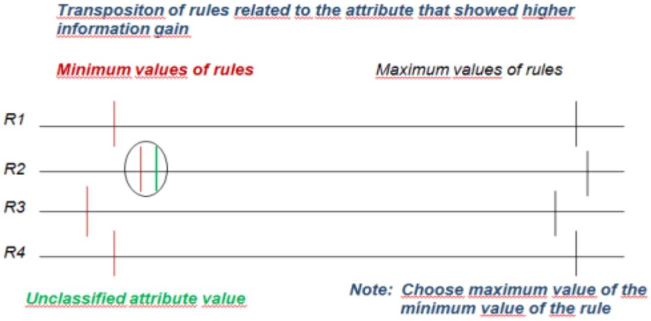

Transpositon of rules related to the attribute that showed higher information gain
Minimum values of rules
Maximum values of rules
R1
R2
R3
R4
Unclassified attribute value
Note: Choose maximum value of the minimum value of the rule

Fig. 2 Selection of the rules to unclassified values

The second option of above presented the best performance.

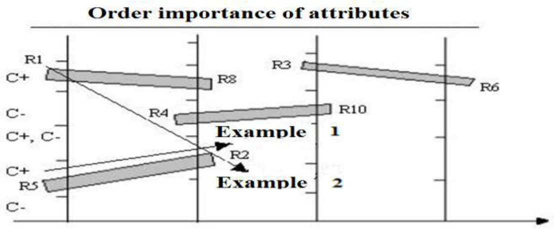


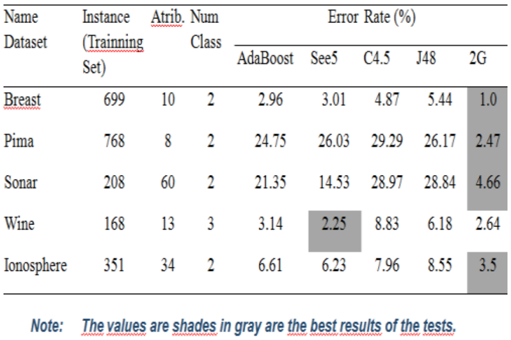
Fig. 3 Classification with “No explicit virtual patterns”

Where in figure 3, class = C+, C- and rules = {R1, …RN}

## 3. RESULTS

Compared with algorithms such as C4.5 [1] [2], See5 [3] and J48 [4], in our case we obtain the biggest degree of information from the attributes in one chance. With regard to the order obtained, it is swapped the first attribute with the second one and the corresponding process of the decision tree building continues. This decision tree is built to provoke bigger data variability. The results obtained from the selection of attributes are similar to those reported in algorithms in [11]. K-fold Cross Validation [12] is used with K = 10. The best results are shown in the gray section of Table III.

Table III- Results comparison



| Name Dataset | Instance (Trainning Set) | Atrib. | Num Class | Error Rate (%) | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | AdaBoost | See5 | C4.5 | J48 | 2G |
| Breast | 699 | 10 | 2 | 2.96 | 3.01 | 4.87 | 5.44 | 1.0 |
| Pima | 768 | 8 | 2 | 24.75 | 26.03 | 29.29 | 26.17 | 2.47 |
| Sonar | 208 | 60 | 2 | 21.35 | 14.53 | 28.97 | 28.84 | 4.66 |
| Wine | 168 | 13 | 3 | 3.14 | 2.25 | 8.83 | 6.18 | 2.64 |
| Ionosphere | 351 | 34 | 2 | 6.61 | 6.23 | 7.96 | 8.55 | 3.5 |

Note: The values are shades in gray are the best results of the tests.

## 4. FUTURE WORK

As future we work in a procedure to reduce the number of rules generated, and is currently working on the implementation of a distribution of solutions based on cross validation, which is testing the selection of the best set of rules using genetic algorithms. This work is still in development, but it is expected that the results will be satisfactory.

## 5. CONCLUSIONS

Among the existing classification techniques, decision trees have proved to be very efficient and accurate enough to generate new knowledge. The 2G algorithm designed with this approach offers original contributions that are not covered by the revised algorithms.

The main contributions of the 2G algorithm are: that instead of trying to reduce the number of values in the process of discretization process, we apply a selection method that includes values than by other methods are ignored, as are the values of overlapping classes, that in this case, we include in the groups that we called “virtual class”, so, we manage to give them representation in the final set of values that are used to select the attributes with the greatest information gain. This has allowed us to maintain better accuracy in the generation of rules, which together with the application of 'no explicit virtual patterns' and additional criteria has led us to have better results in most cases reported in Table III, which includes results for comparison with the algorithms mentioned in the literature.

## REFERENCES

1. Quinlan J., “Induction of Decision Trees”, *Kluwer Academic Publishers, Machine Learning*. (1986).

2.http://www2.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/c4.5/tutorial.html.

3. http://www.rulequest.com/see5-info.html.

4. http://www.cs.waikato.ac.nz/ml/weka/.

5. Bartlett L & Traskin M., “ADABOOST is Consistent”, *Neural Information Processing Systems Conference*. (2006).

6. Díaz M., Fernández M.z & Martínez A., “See5 Algorithm versus Discriminant Analysis. An Application to the Prediction of Insolvency in Spanish Non-life Insurance Companies”, Universidad Complutense de Madrid (2004).

7. Kohavi, R., Li, C.-H., “Oblivious decision trees, graphs, and top-down pruning”, *Fourteenth International Joint Conference on Articial Intelligence*. (2005).

8. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

9. http://www.ailab.si/orange/doc/datasets/breast-cancer-wisconsin.htm.

10. http://archive.ics.uci.edu/ml/datasets.html.

11.http://www.grappa.univille3.fr/~torre/guide.php?id=accueil.

12. Kohavi Ron, *A Study of Cross-Validation and BootStrap for Accuracy Estimation and Model Selection.* International Joint Conference on

Artificial Intelligence. (1995). http://robotics.stanford.edu/users/ronnyk/.

13. Friedman, M. y Kendel, A. *Introduction to Pattern Recognition*. World Scientific (1998). Pp. 1, 3-4, 9-10.
14. Kantardzic, M., *Data Mining: Concepts, Models, Methods, and Algorithms*, Wiley-Interscience, 2003. ISBN:0471228524. pp. 1-56.

15. Brachman R.J. & Anand T., *The Process Of Knowledge Discovery In Databases: A Human-Centered Approach. In Advances In Knowledge Discovery And Data Mining*, eds. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, AAAI Press/The MIT Press, Menlo Park, CA., 1996, pp. 37-57.

16. Comley W., Allison L. & Fitzgibbon J., "Flexible Decision Trees in a General Data-Mining Environment", *Fourth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL-2003)*, Hong Kong

17. Joyanes, L., *Programación orientada a objetos*, Mc Graw Hill, ISBN: 84-481-0585-0, pp. xvii-xix

18. N.J. Nilsson, *Introduction to Machine Learning*. Septiembre, 1996, pp. 81-96