Machine Learning, 6, 231-250 (1991)

The Use of Background Knowledge in Decision Tree Induction

MARLON NÚÑEZ (mnunez@ts.tid.es)
TELEFONICA I+D, C/Emilio Vargas, 6, 28043 Madrid, Spain

Editor: Jaime Carbonell

Abstract. At present, algorithms of the ID3 family are not based on background knowledge. For that reason, most of the time they are neither logical nor understandable to experts. These algorithms cannot perform different types of generalization as others can do (Michalski, 1983; Kodratoff, 1983), nor can they can reduce the cost of classifications. The algorithm presented in this paper tries to generate more logical and understandable decision trees than those generated by ID3-like algorithms; it executes various types of generalization and at the same time reduces the classification cost by means of background knowledge. The background knowledge contains the ISA hierarchy and the measurement cost associated with each attribute. The user can define the degrees of economy and generalization. These data will influence directly the quantity of search that the algorithm must undertake. This algorithm, which is an attribute version of the EG2 method (Núñez, 1988a, 1988b), has been implemented and the results appear in this paper comparing them with other methods.

Keywords. Decision trees, knowledge acquisition, economic induction, background knowledge

1. Introduction

The algorithms of the ID3 family have suffered various limitations with respect to their utilization in many kinds of problems. The principal one being the lack of background knowledge, which limits the process of learning.

One of these limitations consists of the impossibility of applying different types of generalization. In inductive algorithms that use languages based on logic of predicates like AQ11 (Michalski, 1983), INDUCE (Larson & Michalski, 1977) this function is basic and inseparable from the inductive process itself. One of the consequences of this lack of background knowledge is the complexity of the generated decision trees.

Although the decision trees generated by ID3-like algorithms are accurate, they suffer the disadvantage of excessive complexity and are therefore incomprehensible to experts. In order to solve this problem, many systems have been developed. These systems concentrate only on the information contained in the historical data: The PRISM algorithm (Cendrowska, 1989) and the INDUCT algorithm (Gaines, 1989) maximize the information contributed by an attribute-value pair. Previously Quinlan (Quinlan, 1989), Lavrac (Lavrac et al., 1986), Bratko (Bratko et al., 1987) proposed other techniques to generate more understandable and accurate decision trees. Since experts bring additional knowledge to bear while learning decision rules, the proposed method tries to solve this problem using background knowledge.

On the other hand, one of the main difficulties with diagnosis problems is that they may include costly tests. One needs to acquire knowledge that economizes resources, that is to say, the learner should perform economic induction.

When an expert in brain tumors, for instance, receives a patient who suffers a headache, he does not recommend the Scanner as a first diagnostic test, this being the most effective one, because the expert has in mind economic criteria. Therefore the expert asks simple questions and orders other more economic tests in order to discriminate the simple cases, and only recommends such an expensive test for the complex ones. This means that learning methods should take into consideration minimizing costs of the diagnosis in the process of rule induction, in order to be applicable to more problems.

The algorithm presented in this paper undertakes various types of generalization and reduces the classification cost by means of background knowledge. This algorithm, called EG2 (Economic Generalizer 2) (Núñez 1988a, 1988b) constitutes the base of the ALEXIS II program whose results appear in this paper (see also Núñez, 1990). Although this program represents instances in terms of attribute-value pairs, the EG2 algorithm can handle composite objects as training examples. However, this facility is being implemented at the moment and therefore has not yet been proved with real problems. For this reason it is not presented in this paper.

2. Representation and organization in EG2

The EG2 algorithm is an inductive algorithm that generates a decision tree from a set of examples. Beside the training examples, the user can define the ISA hierarchy and the cost of measurement of each attribute, and some data about the degrees of economy and generalization. These data will influence directly on quantity of search that the algorithm must undertake.

2.1. Economic induction

With the goal of constructing a decision tree, the algorithm has a criterion for selecting attributes based on a relationship "cost/benefit." This criterion is more general than the one proposed by Quinlan (1979), based exclusively on the measurement of the quantity of information (ΔI). EG2 will produce a decision tree which will allow the final user to classify his problem and in turn, economize on his resources: time, money, labor, energy, level of danger, etc. The average cost of classification with the decision tree generated by EG2 is less or equal to that obtained using the tree generated by ID3.

The economic criterion applied by EG2 to build a simple decision tree is based on the selection of the attribute with the *least* ICF (Information Cost Function). This function is defined as the ratio between cost and the discrimination efficiency of the attribute, that is to say, its relationship "cost/benefit," as follows:

$$\text{ICF (attribute i)} = \frac{f(\text{cost of attribute i})}{g(\text{discrimination efficiency of attribute i})}$$

The cost concept should be understood, not only as a monetary unit to measure a value of an attribute, but also as a concept that can be quantified (i.e., distance, time, risk, danger level, etc.)

In order to define the discrimination efficiency function, the signal/noise (S/N) concept as a measure of efficiency of a transmission line has been adapted. The analogy of the above concept (S/N), from Information Theory (Shannon, 1971), is the measure of UI/NI of the analyzed attribute, as follows:

$$g(\text{discrimination efficiency}) = \frac{\text{Useful information}}{\text{Non useful information}} = \frac{\text{UI}}{\text{NI}}$$

With a very different purpose, the Prospector system (Duda, 1981) applies this useful concept (the ratio between two opposite terms) to calculate parameters for approximate reasonings, due to its sensitivity.

Since UI + NI = TI (total information),
therefore

$$g(\,) = \left[\frac{\text{UI}}{\text{NI}}\right] = \left[\frac{\text{TI}}{\text{NI}}\right] - 1$$

In order to know the meaning of (TI/NI), we have:

$$\Delta\text{I} = \text{H(TI)} - \text{H(NI)} = \left[\frac{2^{\text{H(TI)}}}{2^{\text{H(NI)}}}\right]$$

Since

$$\text{TI} = 2^{\text{H(TI)}}$$

$$\text{NI} = 2^{\text{H(NI)}}$$

Therefore

$$\Delta\text{I} = \log_2\left[\frac{\text{TI}}{\text{NI}}\right]$$

$$2^{\Delta\text{I}} = \left[\frac{\text{TI}}{\text{NI}}\right]$$

$$2^{\Delta\text{I}} = \left[\frac{\text{UI}}{\text{NI}}\right] + 1 = g(\,) + 1$$

$$2^{\Delta\text{I}} - 1 = g(\,)^1$$

f(cost) is cost + 1, to allow ICF to be finite when cost = 0. Cost should be $\gg$ 1. ICF includes a variable ω to calibrate the factor of economy. The range of ω is [0, 1]: ω = 1 means maximum economy. Thus, the ICF formula is as follows:

$$ICF_i = \frac{(cost_i + 1)^{\omega}}{2^{\Delta I_i} - 1}$$

where:

ΔI_i is the information gain of attribute i
ω is a variable of calibration of this economic criterion

ICF is calculated for each attribute. EG2 selects the attribute with the *least* ICF.

2.2. Kinds of generalization

Beside the "dropping condition" generalization performed by any TDIDT algorithm[2], EG2 performs other kinds of generalization. We next introduce the two additional generalization processes performed by EG2.

2.2.1. Climbing the generalization tree

Climbing the generalization tree has been used in various forms (Kodratoff, 1983; Michalski, 1983; Winston, 1987), and because of its relevance to concept recognition it has been included in EG2. The algorithm generalizes depending on the ISA hierarchies associated with the attributes. For an ISA relation to be acceptable in a decision tree it has to fulfill certain criteria of completeness and consistency. This kind of generalization is illustrated in Figure 1.

2.2.2. Union of values

The algorithm can apply the union of symbolic values, also called "adding alternative" generalization (Michalski, 1983). If a union fulfills certain criteria of consistency and completeness, the system accepts this union. Figure 2 illustrates this process of generalization:

In general terms, the input/output specification of the EG2 algorithm is given in Table 1.

3. EG2 algorithm

Before describing the algorithm, we shall define as *structured attributes* those that have an ISA hierarchy associated with them. The lowest leaf values of this tree are the only observable values. The other values are called *abstract values*. The hierarchies can be "tangled" (Kodratoff et al., 1986), that is to say, an observable value can belong to more than one abstract value (see figure 3).

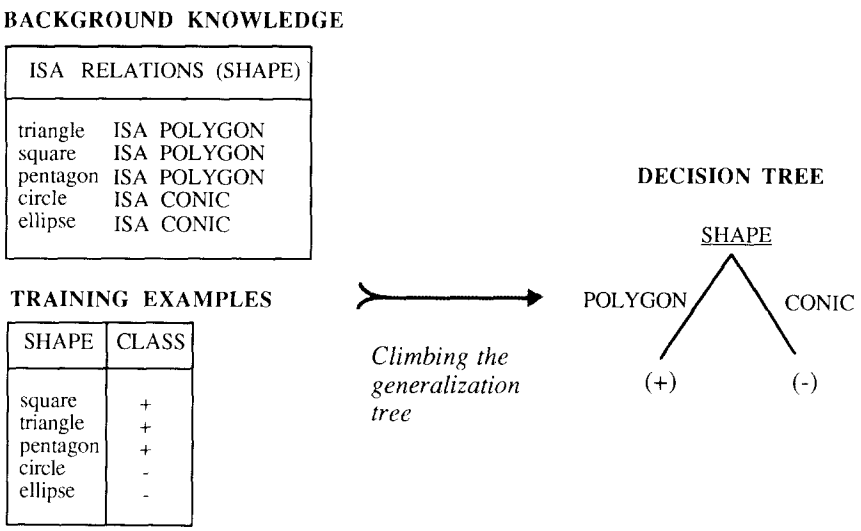


Figure 1. A generalization process using ISA relations.

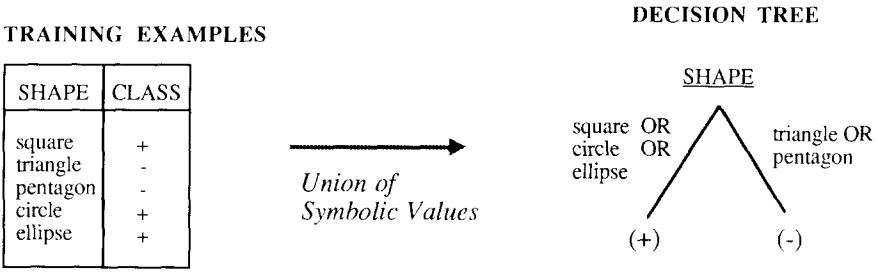


Figure 2. A generalization process using union of symbolic values.

Table 1. Specification of the EG2 algorithm.

GIVEN:

- BACKGROUND KNOWLEDGE:
  ISA hierarchies
  Measuring Costs
- USER DEFINED FACTORS
  Economic factor (ω) [0..1]
  Completeness threshold (ct) [0..1]
- TRAINING INSTANCES:
  Training examples

DETERMINE:

- A decision tree

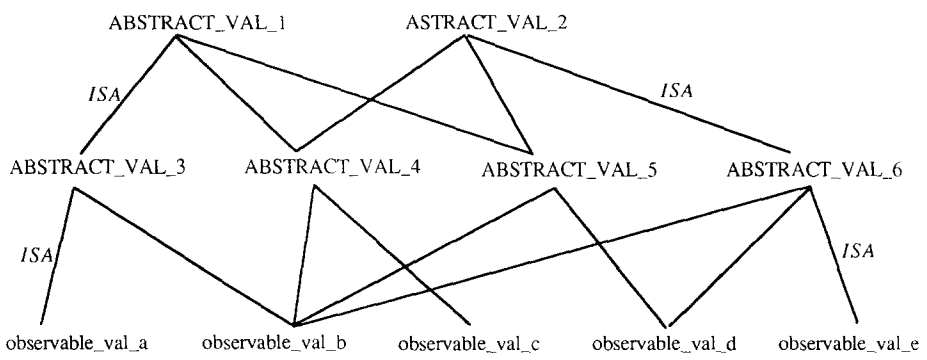


Figure 3. A "tangled" ISA hierarchy.

3.1. Description of the algorithm

EG2 is an amplification of the ID3 algorithm (see Table 2). The algorithm works as follows: EG2 selects the attributes using the economic criterion mentioned above. It then initializes an internal LIST with the more general abstract values and those observable values that do not have abstract values associated. If the selected attribute is structured, it selects only one abstract value according to a certain criterion, explained below. If the selected attribute is not structured, the algorithm selects one of the observable values. Then, EG2 generates a subtree according to this value (abstract or observable). If this subtree is consistent and complete, the algorithm stores it temporarily; the algorithm will try to get a better generalization than the one stored, but if this attempt does not work, it will keep the stored subtree.

In order to get a better generalization, the algorithm tries the best union of abstract values and observable values. Therefore EG2 attempts the former valid value and other values (abstract or observable), and builds a subtree according to this union. This process is repeated until an inconsistency or incompleteness is detected. When that happens, EG2 recalls the last stored subtree. If an observable value cannot be generalized, EG2 generates a subtree according to this observable value (like ID3 does). The process for each selected attribute ends when there are no more values in the internal LIST. Table 2 shows the EG2 algorithm.

Inconsistency is detected when a leaf cannot terminate because there are at least two examples in the subset of examples that are described equally but have different classes.

Incompleteness of a generalization is detected by measuring the proportion of its observable values in all leaves of the subtree below the said generalization. This measure should be greater than a threshold predefined by the user (ct). If each leaf contains each of the values of the generalization, the completeness is 1. In general, this factor is calculated as follows:

$$\text{Completeness factor} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \text{leaf j with the i-th value of the generalization}}{\text{m leaves * n values of the generalization}}$$

$$= \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \mu(i, j)}{m * n}$$

where

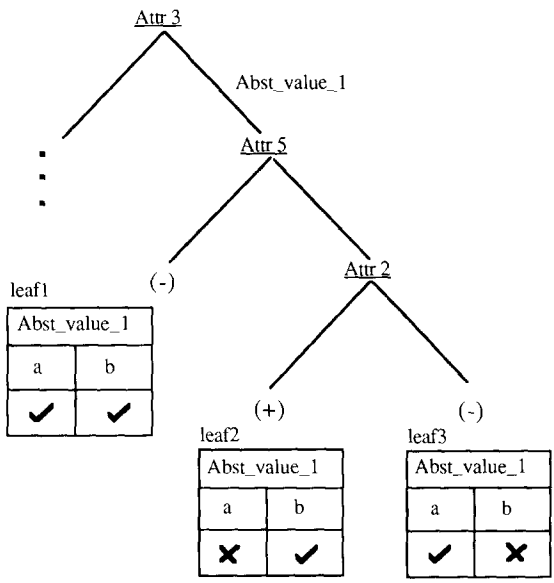
$$\mu(i, j) = \begin{cases} 1 & \text{if the leaf j contains the observable value i} \\ 0 & \text{if the leaf j does not contain the observable value i} \end{cases}$$

Figure 4 illustrates the detection of the incompleteness of an attribute (attribute_3) in one of its generalizations (a, b are observable values of ABST_VALUE_1); we suppose that the user specified a completeness threshold of 0.7.

Table 2. The EG2 algorithm.

```
EG2 (set_of_examples)

IF all the examples have the same class, then label the leaf with this class
ELSE
   Select attribute.  /* in accordance with the economic criterion ICF */
   Initialize LIST with the abstract values of the first level of
      abstraction and the observable values that do not have abstract values associated.
   Let Last_valid_generalization = nil; let Last_valid_subtree = nil
   WHILE LIST is not empty:
      Reorganize LIST
      Construct SUBSET of examples in accordance with:
         Last_valid_generalization OR first element of LIST
      EG2 (subset) /* generate a subtree */
      IF said subtree is consistent and complete
         Let Last_valid_generalization = Last_valid_generalization
                                         OR first element of LIST
         Let Last_valid_subtree = generated subtree
         Destroy generated subtree
         Delete the first element of LIST
         IF LIST is empty
            Add Last_valid_subtree to the final decision tree
            Let Last_valid_generalization = nil
            Let Last_valid_subtree = nil
      ELSE
         Destroy generated subtree
         IF Last_valid_generalization ≠ nil
            Add Last_valid_subtree to the final decision tree
         ELSE
            IF the first element of LIST is an abstract value:
               Decompose said abstract value into observable values
            ELSE
               Inconsistency detected
         Let Last_valid_generalization = nil
         Let Last_valid_subtree = nil
```



$$\text{completeness measure of Abst\_value\_1} = \frac{[\mu(\text{leaf1, a}) + \mu(\text{leaf1, b})] + [\mu(\text{leaf2, a}) + \mu(\text{leaf2, b})] + [\mu(\text{leaf3,a}) + \mu(\text{leaf3,b})]}{\#\text{leaves} * \#\text{ values in the generalization Abst\_value\_1}}$$

$$= \frac{2 + 1 + 1}{2 * 3}$$

$$= 0.66 < ct \qquad \textit{(generalization rejected!)}$$

Figure 4. Illustration of an incompleteness detection.

Figure 5 illustrates the functioning of EG2 with the following exercise. Suppose that the selected attribute is ATTRIBUTE_4 and the background knowledge contains the Abstract Values (AV-1, AV-2 and AV-3) associated to the observable values (a, b, c, d, e and f) of the ATTRIBUTE_4 (note that the value c belongs to two ISA relations):

c, f ISA AV-1
a, b ISA AV-2
e, d, c ISA AV-3

3.2. Search of generalizations

As it was explained above, EG2 tries a UNION between the generalization(s) that has(ve) been previously validated and other values (abstract or observable) not selected before. If a value is selected, it goes to the head of the internal LIST.

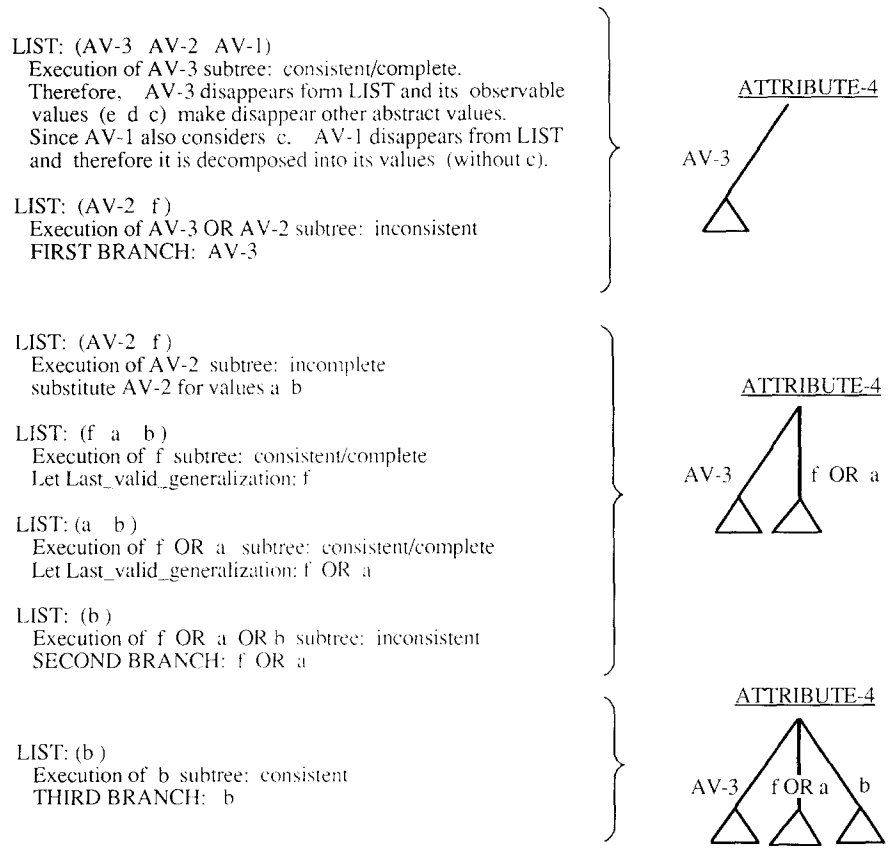


Figure 5. Illustration of a generalization process executed by EG2.

The criteria to select a value (abstract or observable) in order to put it at the head of the internal LIST are the following, in priority order:

1. Abstract values with more observable attributes (they are more general).
2. If there is a conflict (more than one abstract values with the same number of observable values or abstract values mutually exclusive) OR if there are only observable values, then partition the set of examples into subsets, each one according to each possible generalization. Measure the entropy $-\Sigma\ p \cdot \log(p)$ of the class of each subset of examples and select the generalization that produced less entropy. The purpose of this step is to choose the generalization that best classifies the examples.

4. Results

The following examples have been processed by the ALEXIS II program. In the first one, the algorithm applies different types of generalization (climbing the generalization tree and

union of values) and optimizes the classification cost. The other examples show the performance of the economic criterion used by EG2 compared with other approaches in two domains.

4.1. An example

In this example (see Table 3, and Table 4), the user has supplied background data of the problem: his knowledge about the ISA hierarchy and the cost associated to the attributes. He also has supplied the generalization requirements, that is to say, the completeness threshold (ct = 0.6), and the economic factor (ω =1).

Table 3. Background knowledge and training examples of a classification problem.

Background Knowledge			
Attribute	ISA Relations		Measuring Cost
SHAPE	square	ISA POLYGON	$10
	triangle	ISA POLYGON	
	pentagon	ISA POLYGON	
	circle	ISA CONIC	
	ellipse	ISA CONIC	
COLOR	red	ISA PRIMARY	$30
	blue	ISA PRIMARY	
	yellow	ISA PRIMARY	
SIZE	medium	ISA ¬SMALL	$50
	big	ISA ¬SMALL	
	big	ISA ¬MEDIUM	
	small	ISA ¬MEDIUM	
MATERIAL			$100

Table 4. Training examples of a classification problem.

Forma	Color	Size	Material	Class
square	red	big	metal	+
square	blue	small	plastic	+
triangle	yellow	medium	metal	+
triangle	pink	big	leather	−
square	pink	medium	leather	−
circle	red	small	plastic	−
circle	blue	small	metal	−
ellipse	yellow	small	plastic	−
ellipse	blue	big	leather	+
ellipse	pink	medium	wood	+
circle	blue	big	wood	+
triangle	blue	medium	plastic	+

In Table 3 the user employs the operator for logical negation '¬' (i.e., ¬SMALL, ¬MEDIUM). Note also that the value 'pentagon' has been defined in the background knowledge, despite that this observable value is not present in the training set. This situation is usual when the user has divided the historical data into distinct sets for training and testing. ID3 cannot handle this situation. ID3 would classify an instance with shape pentagon as of "UNKNOWN CLASS." EG2 can handle this situation because it will try to generalize square, triangle and pentagon as polygons in order to build the decision tree.

The fact that instances with shape pentagon are not present in the training set will decrease the completeness factor of the generalization POLYGON and therefore it is possible for this generalization to fail. Let us see what happens.

In order to select the first attribute, the algorithm measures ΔI of each attribute, then it applies the economic criterion (ICF) mentioned above.

$$\Delta I = H(TI) - H(NI)$$

$$\Delta I_1 = -\left[\frac{7}{12}\log\frac{7}{12} + \frac{5}{12}\log\frac{5}{12}\right] - \left[\frac{3}{12}\left[\frac{2}{3}\log\frac{2}{3} + \frac{1}{3}\log\frac{1}{3}\right] * 4\right]$$

$$= 0.061$$

$$ICF_1 = \frac{10 + 1}{2^{0.061} - 1} = 252.27$$

Similar analysis gives

$$\Delta I_2 = 0.119 \Rightarrow ICF_2 = 360.54$$

$$\Delta I_3 = 0.167 \Rightarrow ICF_3 = 415.57$$

$$\Delta I_4 = 0.186 \Rightarrow ICF_4 = 733.98$$

Although the attribute MATERIAL has the highest ΔI, SHAPE has the best relation "cost/benefit." EG2 selects attribute 1 (SHAPE) to build the decision tree, and tries to apply the associated abstract value (POLYGON). Then it selects the attribute COLOR, and tries its only abstract value (PRIMARY). Figure 6 shows this induction process.

This subtree is consistent. EG2 calculates the completeness percentage of the generalizations SHAPE = POLYGON and COLOR = PRIMARY as follows:

$$\text{completeness of POLYGON} = \frac{\text{2 values present in first leaf + 2 values present in second leaf}}{\text{2 leaves below branch POLYGON * 3 values for generalization POLYGON}}$$

$$= 0.666 > \text{ct (generalization accepted!)}$$

$$\text{completeness of PRIMARY} = \frac{\text{3 values present in the only leaf}}{\text{1 leaf below branch PRIMARY * 3 values for generalization PRIMARY}}$$

$$= 1.0 > \text{ct (generalization accepted)}$$

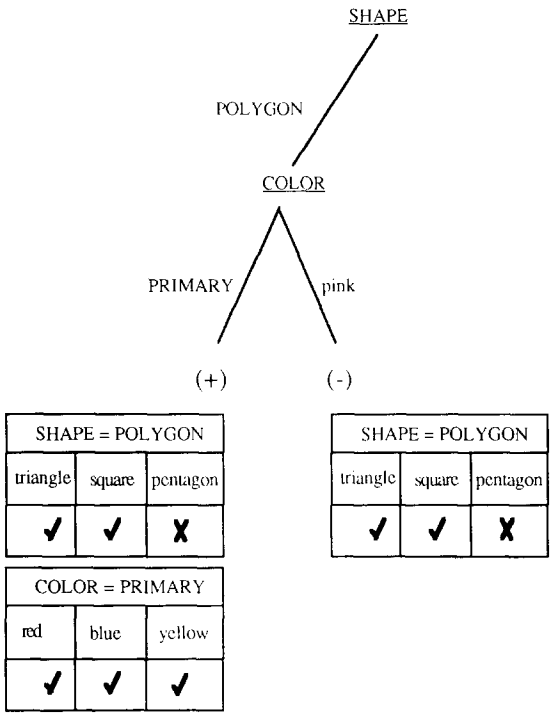


Figure 6. Generalization process of SHAPE = POLYGON and COLOR = PRIMARY.

The next step is to select another abstract value. The only one remaining is SHAPE = CONIC. After constructing this branch, the algorithm selects the attribute SIZE. This attribute has two abstract values (¬MEDIUM and ¬SMALL) with a common observable value 'big.' According to preferences for selecting abstract values, EG2 calculates the entropy of each subset ¬SMALL and ¬MEDIUM (Figure 7). The entropy of ¬SMALL is 0.0 and the entropy of ¬MEDIUM is 0.971, therefore the algorithm selects SIZE = ¬SMALL in order to continue constructing the decision tree. The completeness measure for the generalization ¬SMALL is illustrated in Figure 8.

$$\text{completeness of CONIC} = \frac{\text{2 values present in first leaf + 2 values present in second leaf}}{\text{2 leaves below branch POLYGON * 2 values for generalization POLYGON}}$$

$$= 1.0 > \text{ct (generalization accepted)}$$

$$\text{completeness of } \neg\text{SMALL} = \frac{\text{2 values present in the only leaf}}{\text{1 leaf below branch PRIMARY * 2 values for generalization PRIMARY}}$$

$$= 1.0 > \text{ct (generalization accepted)}$$

The final decision tree is shown in Figure 9. The final classification cost of this tree is $50.

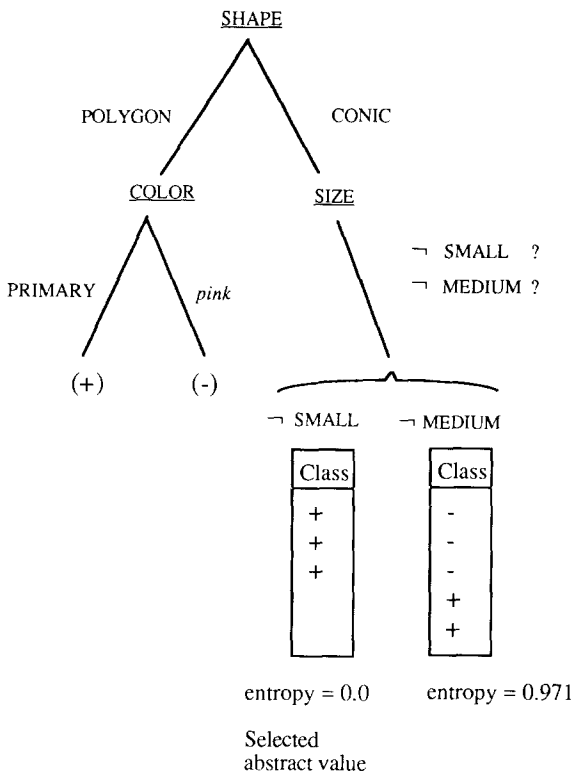


Figure 7. Illustration of the criterion for selecting an abstract value among abstract values mutually exclusive.

Figure 10 shows the decision tree generated by ID3 for the same problem. EG2 can also generate the tree of Figure 10, by inhibiting the economic induction and without applying climbing tree generalization nor union of values.

The trees generated by EG2 are more general than those generated by ID3. The "rules" induced by EG2 are confirmed by more examples than the "rules" induced by ID3, and so the number of leaves of the EG2 tree is less than the number of leaves of the ID3 tree (i.e., in the example, 4 instead of 11). This example shows dramatically that ID3 cannot "see" that polygons have the same classifications or that conic items also have the same classifications in some circumstances. ID3 cannot apply operator for negation (i.e., ¬SMALL) or disjunction.

EG2 can also generate a decision tree with economic optimization ($\omega = 1$), but without considering the background knowledge. This decision tree has a classification cost slightly lower ($45) than that of Figure 9 ($50), in which background knowledge is considered. This is because isolated observable values discriminate better than the generalization of these same values.

Figure 11 shows that the more general a decision tree is, the more costly it is. Thus the more general decision trees (i.e., around ct = 0 which means that almost any generalization is accepted) are more costly than the specific ones (i.e., around ct = 1 or those that

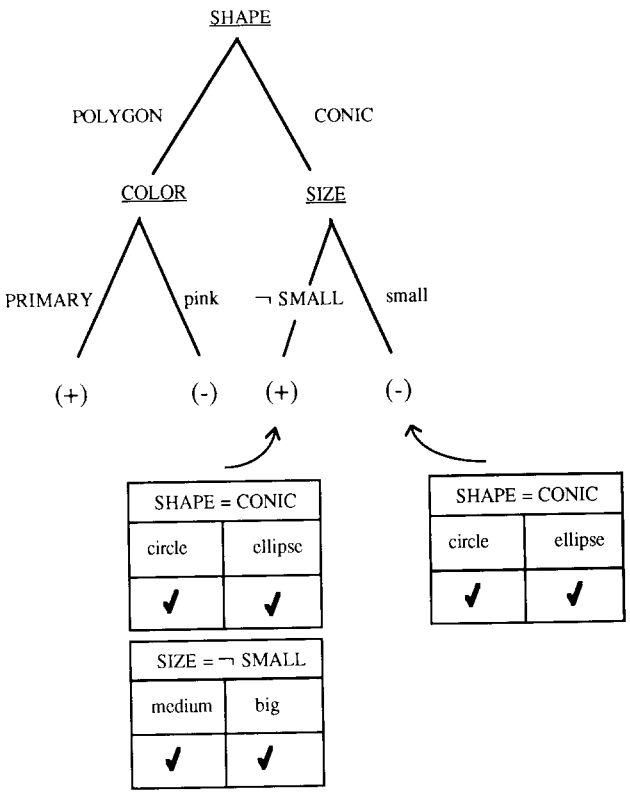


Figure 8. Analysis of the generalizations SHAPE = CONIC and SIZE = ¬SMALL.

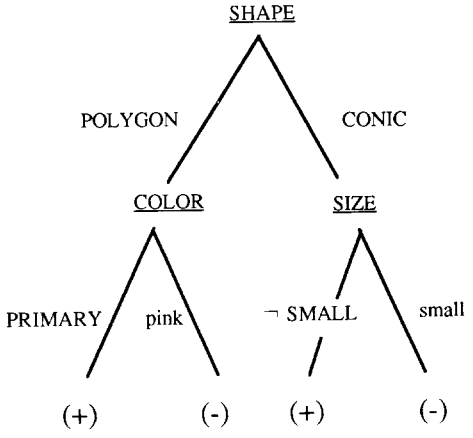


Figure 9. Final decision tree generated by EG2 [ω = 1, ct = 0.6] for the problem of Tables 3 and 4. The average classification cost of the tree is $50.

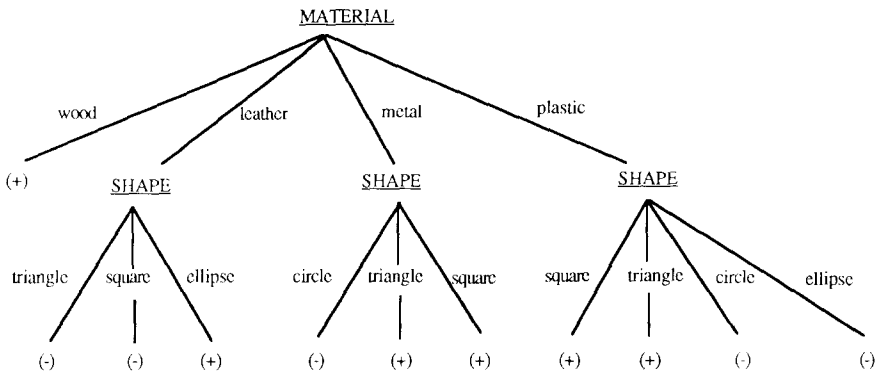


Figure 10. Decision tree generated by ID3 tree or by EG2 without applying economic induction [$\omega = 0$], climbing tree generalization nor union of values for the problem of Tables 3 and 4. Classification cost: $108.

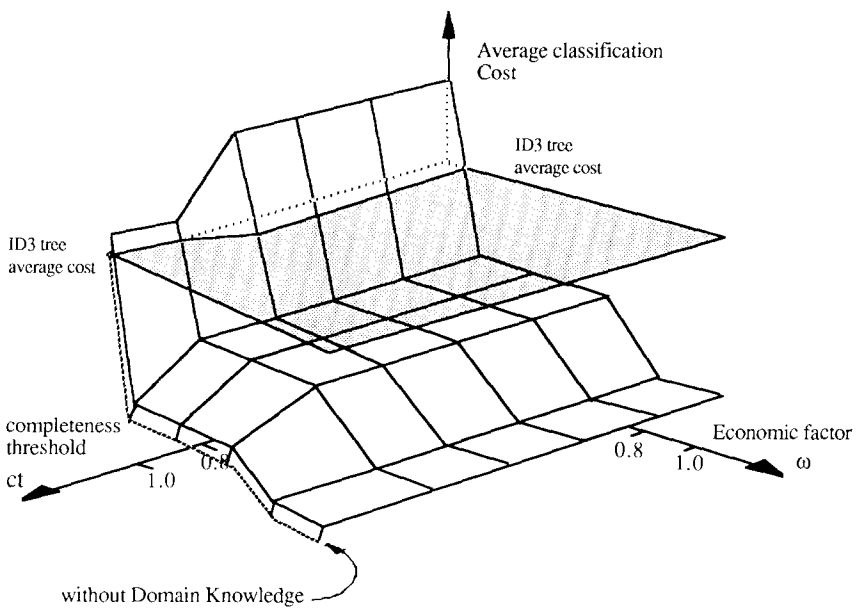


Figure 11. Evolution of the economical performance for any combination of ω and ct (completeness threshold).

do not consider background knowledge). However, a decision tree with generalizations can be more "logical" or easily understood, that is to say, it can best describe a concept. On the other hand, general descriptions can be more precise for testing instances not present in the training set than those less general descriptions (Bratko et al., 1987).

4.2. Performance of the economic criterion of EG2

In order to test the attribute selection criterion of EG2, several domains were analyzed. We begin by showing in detail the results obtained in one of these domains (medicine). Then we shall show a comparison table contrasting the economical performance of three attribute selection criteria for building decision trees in two domains.

4.2.1. Gynecological diagnosis problem

The learning set of the gynecology diagnosis problem is constituted of 352 cases of women with "amenorrhea" (failure in menses). Each case is described by six attributes and a class. Five classes have been considered, each one indicates an amenorrhea cause: pregnancy, menopause, psychological cause, low grade hormonal failure and "other pathologies." If the program cannot give a reliable diagnosis, it classifies the case in "other pathologies," because this application was thought to discriminate the routine, which are the majority of cases, from the complex ones.

Associated with each attribute is an associated measurement cost. The most costly and also effective attribute is the Hormonal Profile Test which classifies every case (see Table 5). Another attribute with a lower cost is the pregnancy test, which can only discriminate the class "pregnancy." The other attributes had no cost. The selection of the first attribute is shown in the following table.

The discrimination efficiency ($2^{\Delta I} - 1$) of the Hormonal Profile Test is very high, but also very expensive; therefore the ICF is high. On the other hand, being the third in discrimination efficiency, "age" has cost = 0.0 and so the ICF function is very low; and thus "age" is selected as the first attribute.

The learning set was tested on ID3 and EG2 [$\omega = 1$]. The generated trees were quite different. ID3 generated a very small tree with just one node in which the selected attribute was Hormonal Profile Test, which discriminated every case. EG2 [$\omega = 1$] generated a larger tree whose first selected attribute is "age" (with symbolic values). It is not a coincidence that one datum every gynecologist needs to know for this particular problem is the age of the patient.

To assess the economic performance of the EG2 economic criterion, the variable ω was set to 1 and to 0 (ID3). The average cost of each diagnosis using ID3 (or EG2 [$\omega = 0$]) was 8000 units, while it was 3840 units using EG2 [$\omega = 1$], realizing a 52% improvement.

Table 5. Measures of ICF for selecting the first attribute in the gynecology problem.

Attribute	ΔI	$2^{\Delta I} - 1$	Cost	ICF
Menstruation delay	0.108	0.078	0	12.82
Age	0.463	0.379	0	2.63
Early Pregnancy Disorders	0.081	0.058	0	17.24
β.HCG Pregnancy Test	0.930	0.906	2000	2207.39
Hormonal Profile Test	3.142	7.83	8000	1171.44
Coitus Last Period (Yes/No)	0.365	0.288	0	3.47

Figures 12 and 13 show the general performance (average classification cost and average depth) using EG2, depending on the economy calibrator (ω) and the number of attributes (n). The average classification cost for $0 < \omega < 1$ is greater than the average classification cost for $\omega = 1$.

4.2.2. Comparative results

Tan and Schlimmer[3] (1989, 1990) proposed another criterion for selecting attributes considering costs, similar to the one used in the EG2 algorithm (Nuñez, 1988a, 1988b). According to this criterion, the algorithm selects the attribute with the highest ΔI^2/cost. This criterion also constructs decision trees with less classification cost than ID3.

To compare the three attribute selection criteria (ID3, EG2, and Tan and Schlimmer's), two domains were analyzed: The gynecology domain (problem 'a') analyzed above and another for classfying flags (Forsyth & Rada, 1986), for 10 different cost situations. Table 6 shows the economic performance of the three attribute selection criteria in two domains for 10 different assignments of measuring attribute costs.

The percentages show how much less expensive each decision tree is by using an evaluation function (Tan and Schlimmer's and EG2) compared to the tree generated using ID3. Table 6 shows that the decision trees generated by EG2 are cheaper than or equal to the decision trees generated by the Tan and Schlimmer approach.

This approach is closer to ID3 than to EG2 [$\omega = 1$]. The average classification cost of the Tan and Schlimmer approach is between EG2 [$\omega = 1$] and ID3, and the average depth of the generated tree is also between EG2 [$\omega = 1$] and ID3.

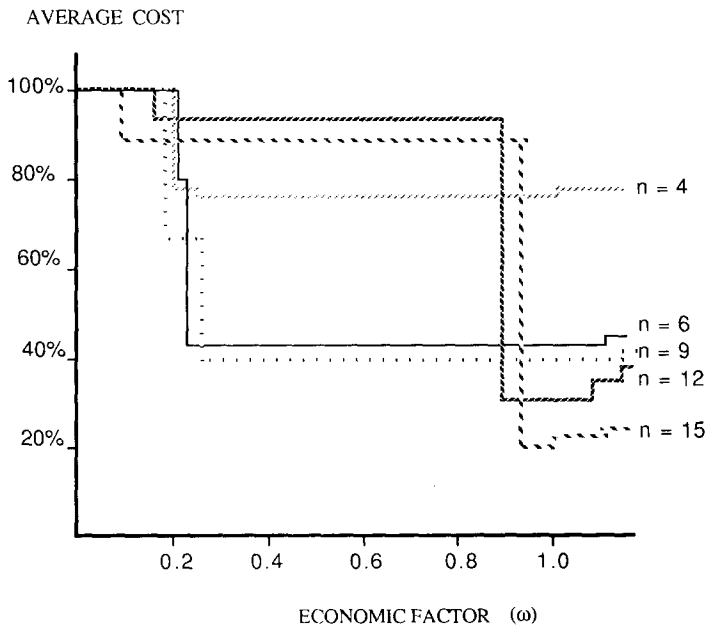


Figure 12. Average classification cost as a function of ω, for several values of n (100% cost: ID3 tree).

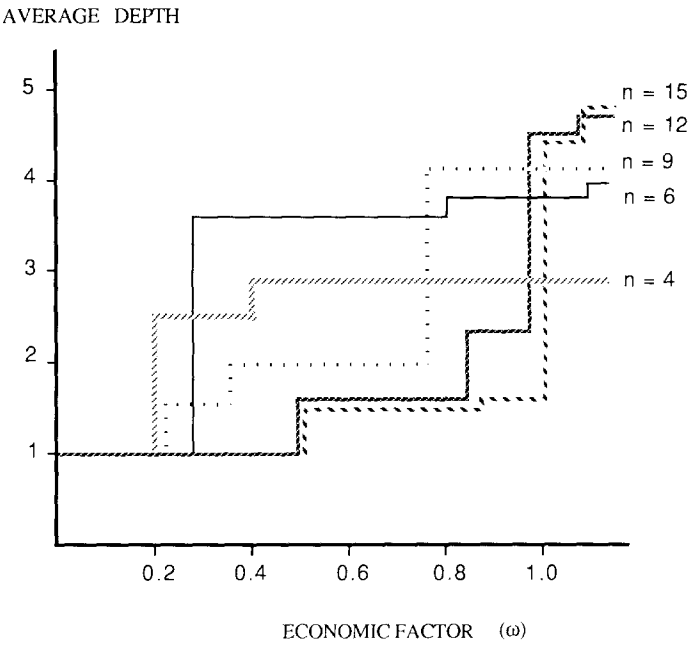


Figure 13. Average depth as a function of ω, for several values of n (depth 1: ID3 tree).

Table 6. Economic performance of the three criteria for selecting attributes.

Problem	ID3	Tan & Schlimmer's	EG2 [ω = 1]
Cost situation a.1	8000	4532 (56%)	3840 (48%)
Cost situation a.2	4362	3576 (82%)	2312 (53%)
Cost situation a.3	1501	1201 (80%)	930 (62%)
Cost situation a.4	4157	3907 (94%)	3907 (94%)
Cost situation a.5	119	98 (82%)	86 (72%)
Cost situation a.6	73	55 (75%)	46 (63%)
Cost situation b.1	211	141 (67%)	84 (40%)
Cost situation b.2	14026	12489 (89%)	8365 (60%)
Cost situation b.3	27092	18029 (66%)	14903 (55%)
Cost situation b.4	35484	30163 (85%)	27679 (78%)
Average percentage	(100%)	(77%)	(62%)

5. Conclusion

A method of inductive learning that focuses mainly on economy of resources has been presented. Economy is a very important part of common-sense reasoning and should be included in learning systems.

Background knowledge is a necessary component of inductive learning, as has been proved in previous systems (Michalski, 1983; Kodratoff, 1986). The method described here uses background knowledge to provide information for performing economic induction and to provide constraints and guidance for building more general and suitable decision trees.

Acknowledgments

I am grateful to Jaime Carbonell for his useful comments and suggestions. I would like to thank Luis Esteban Garcia from ETSIT (Universidad Politécnica de Madrid) who programmed ALEXIS II, to Juan Pazos for his encouragement and comments, to Zandra Montañez and Luz A. Uribe M.D. from Clinica de la Concepción for formalizing and providing data for the Gynecology domain. I am also grateful to Pedro L. Muñoz for his useful comments about the organization of this paper.

Notes

1. It is redundant to calculate the exponential of a logarithmic formula ($2^{\Delta I}$); therefore $g(\ )$ can also be calculated as follows:

$$g(\ ) = \frac{TI}{NI} - 1$$

Where:

$$TI = 2^{H(TI)} = 2^{-\sum_i p(i)\cdot\log_2(p(i))} = \prod_i p(i)^{-p(i)}$$

$$NI = 2^{H(NI)} = 2^{\sum_r P(r)\cdot[-\sum_q p(q,r).\log_2(q,r)]} = \prod_r [\prod_q p(q,r)^{-p(q,r)}]^{P(r)}$$

2. Top Down Induction of Decision Tree.
3. Tan and Schlimmer built a robot system (CSL) that learns to sense unknown objects and select grasping procedures for them. This Cost-Sensitive Learning System analyses which sensors to use, where to use them, and how to generate an inexpensive control procedure to accomplish its task.

References

Bratko, I., & Kononenko, I. (1987). Learning rules from incomplete and noisy data. In B. Phelps (Ed.), *Interactions in artificial intelligence and statistical methods*. Hampshire: Technical Press.

Cendrowska, J. (1989). PRISM: An algorithm for inducing modular rules. In B. Gaines, & J. Boose (Eds.), *Knowledge acquisition for knowledge based systems* (Vol. 1). Academic Press.

Duda, R. (1981). Model design in the prospector consultant system for mineral exploration. In D. Michie (Ed.), *Expert systems in the microelectronic age*. Edinburgh, Scotland: Edinburgh University Press.

Forsyth, R., & Rada, R. (1986). *Machine learning applications in expert systems and information retrieval*. West Sussex, England: Ellis Horwood Publishers.

Gaines, B. (1989). Knowledge acquisition: The continuum linking machine learning and expertise transfer. In J. Boose, B. Gaines, & J. Ganascia (Eds.), *Proceedings of the EKAW 89*. Paris.

Hunt, E.B., Marin, J., & Stone, P.J. (1966). Experiments in induction. New York: Academic Press.

Kodratoff, Y., & Ganascia, J. (1986). Improving the generalization step. In R.S. Michalski, J.G. Carbonell, & T.M. Mitchell (Eds.), *Machine learning* (Vol. 2), Los Altos, CA: Morgan Kaufmann.

Larson, D., & Michalski, R. (1977). Inductive inference of VL decision rules. *Proceedings of the WPDIS-77*. Sigart.

Lavrac, N., Mozetic, I., & Kononenko, I. (1986). An experimental comparison of two learning programs in three medical domains. *Proceedings of ISSEK Workshop 1986*. Glasgow: Turing Institute.

Michalski, R. (1983). Theory and methodology of inductive learning. In R.S. Michalski, J.G. Carbonell, & T.M. Mitchell (Eds.), *Machine learning* (Vol. 1). Los Altos, CA: Morgan Kaufmann.

Núñez, M. (1988a). El metodo de aprendizaje EG2: Una aplicacion de conociminento de base a ejemplos estructurados. *Master thesis in knowledge engineering*, Universidad Politecnica de Madrid.

Núñez, M. (1988b). Economic induction: A case study. In D. Sleeman (Ed.), *Proceedings of the Third European Working Session on Learning*. London: Pitman Publishing.

Núñez, M. (1990). Decision tree induction using domain knowledge. In B. Wielinga, J. Boose, B. Gaines, G. Schreiber, & M. van Someren (Eds.), *Current trends in knowledge acquisition*. Amsterdam, The Netherlands: IOS Press.

Quinlan, J.R. (1979). Discovering rules by induction from large collection of examples. In D. Michie (Ed.), *Expert systems in the microelectronic age*. Edinburgh University Press.

Quinlan, J.R. (1989). Simplifying decision trees. In B. Gaines, & J. Boose (Eds.), *Knowledge acquisition for knowledge based systems* (Vol. 1). Academic Press.

Shannon, C.E., & Warren, W. (1971). *The mathematical theory of communication*. University of Illinois Press.

Tan, M., & Schlimmer, J. (1989). Cost-sensitive concept learning of sensor use in approach and recognition. *Proceedings of the Sixth International Workshop on Machine Learning*. Ithaca, New York.

Tan, M., & Schlimmer, J. (1990). CSL: A cost-sensitive learning system for sensing and grasping objects. *IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio.

Winston, P.H. (1977). *Artificial intelligence*. Addison-Wesley.