COMBINING SYMBOLIC LEARNING TECHNIQUES AND STATISTICAL REGRESSION ANALYSIS

CARLO BERZUINI

Dipartimento di Informatica e Sistemistica. Universita' di Pavia.
Via Abbiategrasso 209. 27100 PAVIA (ITALY)

Abstract

This paper discusses relationships between statistical modeling techniques and symbolic learning from examples, and indicates types of learning problem where a combined viewpoint may be very helpful. A novel computational approach is proposed which combines statistical modeling with a transformation procedure which maps the statistical model onto logical decision rules for the sake of domain experts' intuitions. The proposed algorithm is illustrated by working through a simple but challenging case-study on learning prognostic rules from clinical observational data.

1. INTRODUCTION

Noise, uncertainty and incomplete information can severely degrade the quality of rules generated by a system for inductive learning from examples. Although several algorithms have been developed which attempt to deal with noisy domains, still the following remain crucial issues.

Probabilistic vs. deterministic concept expression. Because of uncertainty, learning must often be done, rather than in terms of few "crisp" categories, in terms of a smooth gradation of multiple categories representing narrow ranges of probability. Exg. if we want to recognize patients affected by a given disease from normal ones, on the basis of some attributes, two categories (normal and diseased) may be unsufficient. It may well be better to define, and characterize by the value of the attributes, multiple categories at different degrees of risk of disease.

Managing noise. When there is noise, arising from errors in the description of attributes or classes, or some inherent uncertainty in the domain, it may be the case that two examples share the same attribute values and have different class values ("clash").

In this paper we propose a framework in which a well-known statistical technique, *regression analysis,* and symbolic learning techniques may efficaciously interact in order to solve with renewed efficiency the problems above. As an example, reconsider the problem of discriminating normal and diseased patients, on the basis of, say, two attributes X_1 and X_2. On the basis of a training sample of normal and diseased patients, we can estimate the parameters of a logistic regression model $log\,(p/(1-p))=\phi(x_1,x_2)=\beta_0+\beta_1 x_1+\beta_2 x_2$, where p is the posterior probability of disease, and then define by inequality constraints on $\phi(x_1,x_2)$ a smooth gradation of "risk categories" characterized by small ranges of p.

The proposed approach, which combines regression analysis and inductive learning heuristics, has two phases. In the first, regression analysis is exploited as a "numerical engine" for selecting and estimating the parameters of a statistical model which adequately reflects the "true" predictive relationships suggested by the data. In the second phase, a novel computational procedure "maps" the algebraic constraints upon attributes implied by the statistical model into symbolic concept descriptions, structured as binary trees or decision rules, for the sake of psychological meaningfulness.

In order to obtain a natural-to-understand final product of the learning, *loss* of predictive efficiency with respect to the regression model must be traded-off for "simplicity". This implies searching among a large set of logical descriptions "reasonably" consistent with the statistical model.

2. REGRESSION ANALYSIS

In regression analysis, the set of examples, or *learning sample,* consists of N pairs of observations $(Y_i,\mathbf{X}_i)$, where each $\mathbf{X}_i$ is the p-dimensional vector of attributes of the i^{th} example, and Y_i is a real-valued number, called **response.** Examples of response are: survival time, probability of belonging to a diagnostic category, a.s.o.

The problem tackled by regression consists in using the learning sample to acquire knowledge useful for at least one of the following aims: *(a)* obtain $\hat{Y}$, i.e. the prediction of the value of Y corresponding to future measured $\mathbf{X}$-vectors as accurately as possible, and *(b)* understand the structural relationships between Y and attributes in $\mathbf{X}$.

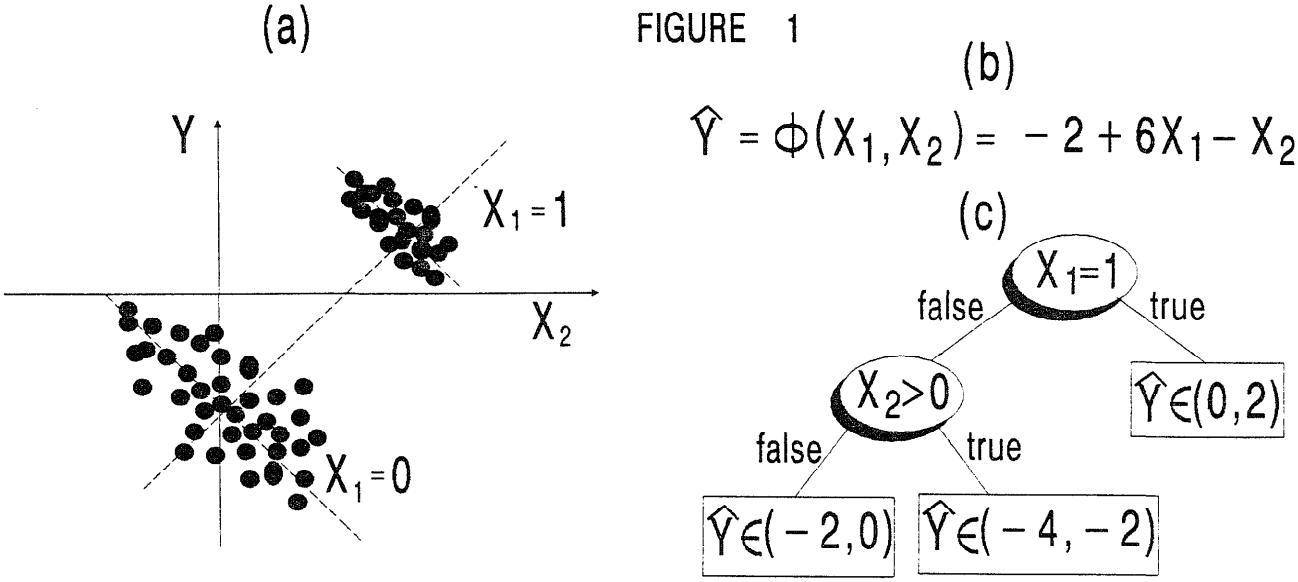
The regression model represents $Y_1,\ldots,Y_N$ by random variables with, for some appropriate function g and **predictor function** ϕ:

$$\hat{Y}_i = E(Y_i) = g(\phi(\mathbf{X}_i;\beta),\alpha) \qquad (1)$$

where $\hat{Y}=E(Y_i)$ is the **predicted**, or **expected** value of Y_i, α and β being vectors of unknown parameters. Regression analysis provides

This work was partially supported by C.N.R. grant no. 87.01829, MPI40% aod MPI60%

FIGURE 1



procedures for estimating such unknown parameters from the learning sample by likelihood maximization. To allow estimation of parameters, the regression model has to be completed by explicitly modeling the "noise" on the data (which, importantly, is something AI procedures ignore). This amounts to specifying the probabilistic distribution of the Y's.

The predictor function $\phi(\mathbf{X};\beta)$ is the minimal summary of the measurement vector $\mathbf{X}$ sufficient to determine Y. That is to say that it "tells the whole story" about the *structure* of the relationships between Y and $\mathbf{X}$. The form of g and α parameters are important, but unrelated to stable notions in the human memory such as the interactions between X variables in predicting Y.

Note that (1) is in general a nonlinear model. An interesting class of models, called *generalized linear models,* has ϕ linear in the parameters β.

As an example, consider the scatter diagram in Fig. 1*a* summarizing a fictitious data set where each sample individual is characterized by two continuous variables (Y, X_2) and a binary one (X_1). By regressing Y on (X_1, X_2), we obtain the model reported in Fig. 1*b*.

3. BINARY REGRESSION TREES (BRT)

The regression model in Fig. 1*b* can be approximated through the *Binary Regression Tree* (BRT), in Fig.1*c*. Through a sequence of binary splits based on the value of explanatory variables, the BRT partitions the entire population, represented by the root of the tree, into a hierarchically organized set of subpopulations, represented by the other nodes of the tree. Each edge of the tree is labeled by an assignment of *true* or *false* to a **binary** variable, which sometimes is an attribute originally coded as true or false, and sometimes is a logical expression of the form ($x > x'$), obtained by placing a cut-off point x' on the continuous range of an attribute X.

For a given node, the unique path from the root to it determines an assignment of values *(true, false)* to a subset of the binary variables. The logical conjunction of these assignments defines the subpopulation associated to that node. If these assignments involve the entire set of binary variables, they determine the value of the predictor function uniquely, so that a constant expected value for the response variable is associated to that node. If the assignment is incomplete, the predictor function in that node is constrained within an *interval.* Intervals associated to a set of nodes at a given depth of the tree may be *not* disjoint. Descriptions by means of a BRT are more explicit, informative, and directly usable than the regression equation alone, though somewhat less precise in predicting the response.

4. THE PROPOSED APPROACH

We propose an approach by which **a BRT, or a decision rule, is generated as an explanation of a previously obtained regression model.** This amounts to finding *simply* described classes, representing groups of examples with *homogeneous predicted response.*

We point out two assumptions implicit in the approach. First, that domain experts like logical class descriptions of *conjunctive* type. Second, that they like classes that can be linearly "ranked", in the sense that they correspond to *disjoint* intervals of response values.

The approach consists of three phases:

1. select a parsimonious predictor function $\phi(\mathbf{X})$ from the learning sample by means of regression analysis.

2 construct a complete BRT representing the selected predictor function

3. simplify the BRT so that the final BRT indicates a reasonable number of logically defined classes

Let's examine the three steps in more detail.

4.1. Selecting a predictor function

The aim of predictor selection is to achieve an *economical* form of $\phi(\mathbf{X})$ consistent with the data of the learning sample, by reducing the number of X −attributes included in $\phi(\mathbf{X})$. The need for such simplification arises particularly when the number of predictors is large, and there is an *a priori* suspicion that many of them are measuring essentially equivalent things, and/or some of them may be totally irrelevant to Y. Not only does a "parsimonious" model enable one to think better about the structure of predictor–response relationships, but prediction of response of new cases is more precise if unnecessary terms are excluded from the predictor function. However, it's worth while exploring the consequences of leaving in the model more terms than a strict statistical significance criterion would indicate.

There is a large literature about this topic: see for example [Aitkin,1978]. So we won't further discuss this aspect of the proposed methodology.

4.2. Growing a complete BRT.

A BRT such as the one in Fig. 1c is generated, whose leaves correspond to subpopulations within which the predictor function is constant. This kind of BRT, namely where each leave corresponds to a complete assignment of value to all attributes included in the predictor function, is called **complete.** Usually the complete BRT is too complicated to convey a simple interpretation of the data. Therefore, a "simplification" phase, described in the following, is needed.

4.3. Simplifying the BRT

The complete BRT is submitted to a simplification algorithm, whose output is a pruned tree, with a smaller number of leaves. Each of these leaves generally *constrains* the predictor function to lie within an *interval.* When there are overlapping intervals, suitable *exception terms* are added to the logical description of some leaves, so that final descriptions individuate classes of examples with well–separated response predictions. This approach privileges "cue validity" with respect to "category validity".

The simplification procedure may proceed further with an *amalgamation* phase. This is fusing pairs of leaves if corresponding response predictions are not well–separated (either by statistical or subjective criteria). If the fused leaves have different "parent" node, the tree becomes a *lattice,* i.e. presents multiple–connections between nodes. This introduces *disjunctions* in the class descriptions.

The following section intends to convey the basic idea of the whole approach by illustrating it upon a clinical case–study.

5. AN APPLICATION

In a survival study interest centres on a group or groups of patients for each of whom there is a defined point event which we will call *failure,* occurring after a length of time called *the failure time.* Values are available for each patient of clinical attributes *a priori* thought to be related to failure time.

As an example we will consider a sample set of data concerning Myelofibrosis with Myeloid Metaplasia (MMM), a chronic myeloproliferative disorder. The learning sample comprised 138 patients with MMM consecutively seen in the Department of Internal Medicine and Medical Therapy of the University of Pavia from 1973 to 1985.

There were 20 attributes for each case, including haematological laboratory tests, histological examination and results from tracer studies. Most attributes were of continuous type, others took values on an ill–defined quantitative scale, or were of binary type (exg. sex).

Time from MMM diagnosis to death was regarded as failure time for our sample cases. Our aim was defining a prognostic classification of MMM into meaningful classes with different expected failure time.

While a linear combination of attributes could efficiently explain the statistical variability of failure, nevertheless we wanted results of the data analysis to be expressed in a more "natural" and better structured form, so as to allow clinical experts to better confront them with their personal knowledge.

A method used by many clinicians is to dichotomize according to survival or nonsurvival at a critical period such as five years. In case of dichotomization, learning from pre–classified examples can be used. This approach is often quite unsatisfactory, for the following reasons. First: concentration on a single time point of the survival experience necessarily wastes some information. The critical time threshold ought itself to be determined in such a way that wasted information is minimized. Second, allowance should be made for more than two disjoint intervals over the range of failure times. But how can their number an location be optimized ?

This learning problem is further complicated from the frequent difficulties encountered in obtaining relevant data. In particular, some patients of the learning sample *may not have been observed for the full time to failure.* As a matter of fact, for only 60 of our 137 sample cases death was observed. For the remaining 77 cases only a "censored" survival time was available, that is we only knew that it was higher than a certain value.

The regression step

We used Cox's regression model [Cox,1972] which allows correlating *censored* failure time observations with a set of (mixed–type) attributes. This model fully characterizes an individual for the purpose of predicting failure time by a linear combination $\phi(\mathbf{X})$ of the attributes, which has the

meaning of a relative risk. In fact, Cox's model assumes that the ratio of the instantaneous risks of death for any two individuals A, B with measure ment vectors $\mathbf{X}_A$ and $\mathbf{X}_B$ respectively, is constant over time, and given by $\log(\phi(\mathbf{X}_A)-\phi(\mathbf{X}_B))$. Based on MMM data, we performed a hierarchical set of likelihood-ratio tests to select terms for inclusion in $\phi(\mathbf{X})$. Then we dichotomized continuous attributes by choosing optimal cut-offs on a likelihood maximization basis. The final form of the predictor function was:

$$\phi(A,H,C,T) = -6.51+1.9\,A+0.85\,H+4.8\,C+3.9\,\neg C\,T \quad (2)$$

where: A = (Age > 45 *yrs*), H = (Hb < 13 *g/dl*), C = (Cellularity = aplastic), T = (TEIT < 200).

Rather than restricting himself/herself to patterns of additive composition of attribute effects, one ought better to try in ϕ patterns of *interaction* among attributes. The interaction term $(\neg CT)$, for example, implies that the "effect" of having $TEIT < 200$ is to be taken into account only if the cellularity is not aplastic.

Growing the complete BRT

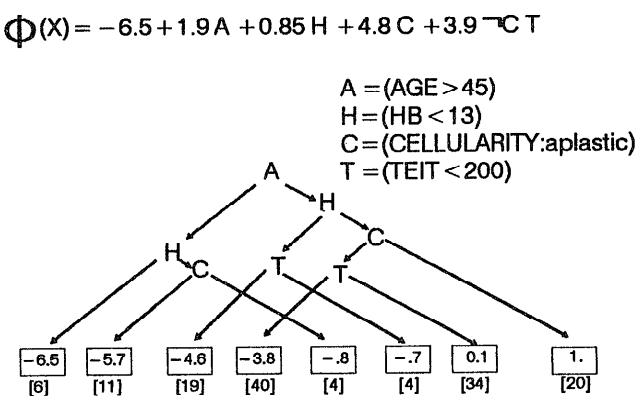


FIGURE 2

Fig. 2 shows the complete BRT grown from $\phi(A,H,C,T)$. Attributes to which the domain expert attached more importance were used for top-level splittings.

Each leave of the BRT contains in the box the value of $\phi(A,H,C,T)$, and in square brackets the number of sample cases associated to it. The leaves are ordered from left to right in the figure according to increasing value of the predictor function. The BRT is unbalanced, because the leaves with an empty set of sample cases were pruned.

The simplification step

The BRT shown in Fig. 2 was simplified by means of the algorithm described in sec.7, and then the pair of leaves corresponding to the highest risk were amalgamated since the domain expert didn't perceive that they represent substantially different classes. The result of this process is in Fig.3. The leaves correspond to 4 risk classes characterized by

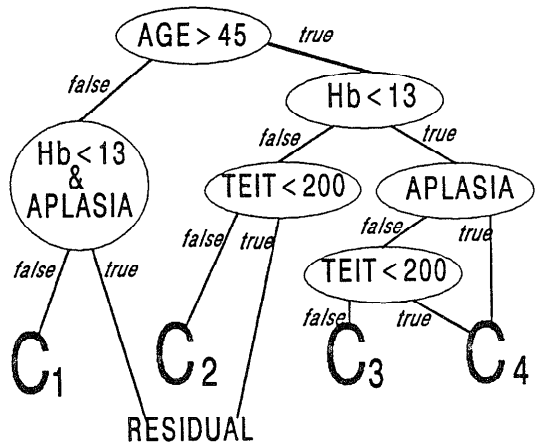


FIGURE 3

intervals on $\hat{Y}$ completely consistent with the original predictor function (2). The "price" to be paid for the simplification is having a RESIDUAL subpopulation of "unclassified" patients. The algorithm in sec. 7 allows minimizing the residual. In fact, only 8 of our 138 cases fell in it.

A clinical expert translated the simplified tree:

"*Four risk-classes $C_1 \cdots C_4$ in order of increasing risk were singled out. C_1 was entirely formed by all patients with age < 45, except a very small portion of them that had red cell aplasia. C_2 was formed by all and only cases above-45 with a very favourable pattern of erythropoiesis, as indicated by absence of anemia and marked expansion of erythropoiesis. C_3 was formed by anemic patients above-45 without severe erythroid failure. C_4 was formed by patients with anemia caused to severe erythroid failure, this latter being indicated by presence of red cell aplasia or of extremely reduced TEIT.*"

6. RELATIONS TO PREVIOUS WORK.

A number of induction algorithms have been developed for dealing with noisy domains and avoiding overfitting. PLS1 [Rendell,1987], for example, is capable of dealing with classes defined on a "probabilistic continuum". [Quinlan, 1983] and [Breiman, 1984] propose *recursive splitting* (RS) algorithms in order to build a *decision tree*, and propose pruning an already created decision tree to obtain an "honest-sized" tree.

Our proposed algorithm may be compared with RS algorithms with relation to a number of issues.

"*Global*" *rather than* "*local*" *tests.* In a RS algorithm, the criterion for selecting a split at a node takes into account only the limited portion of the data represented by that node, while in a regression model each parameter summarizes the effect of an attribute over the *whole* learning sample. As a consequence, our approach is more efficient in managing statistical power in the data, it doesn't easily "loose breath" after a few splits due to the shrinking of the subsets, and doesn't require a stopping criterion.

Stability. RS is known to produce very different results depending on which attribute is chosen for the first split. In RS the splitting criterion doesn't reward splittings in terms of the continued growth of the tree. This means unstability and suboptimality. Model selection in regression is much more stable.

Other advantages of our approach concern the possibility of taking into account "confounding" variables, and of dealing with particular forms of incomplete information.

7. SIMPLIFICATION ALGORITHM

We now formalize how the complete regression tree is simplified and then used to derive logical class descriptions.

We begin with some straightforward definitions.

Let Θ be a complete BRT representing a predictor function $\phi(\mathbf{X})$.

DEFINITION 1. Two nodes of Θ are said to be **independent** when none of them is successor of the other one.

DEFINITION 2. A set of independent nodes is **complete** when each leave of Θ coincides with, or is successor of, *exactly one of them.*

DEFINITION 3. A complete set of independent nodes of Θ, linearly ordered to form a sequence $(N_1, \ldots, N_k)$, is called *I-chain.*

Many I-chains can usually be defined on a BRT. The first and last nodes of the chain are called *root* and *sink* of the chain, respectively.

Let $l_1, \ldots, l_n$ denote the leaves of Θ. If $\phi(l_i)$ denotes the value of the predictor function associated to the generic l_i, we assume for simplicity that leaves can be strictly ordered according to ϕ, and that they are indexed so that:

$$\phi(l_{i+1}) > \phi(l_i) \qquad 1 \leq i \leq n-1 \qquad (3)$$

For a generic internal node N_i of Θ, let $L(N_i)$ denote the set of leaves which are descendants of N_i.

DEFINITION 4. Given two independent nodes N_i and N_j of Θ, the expression :

$$L(N_j) > L(N_i) \qquad (4)$$

means that for any $l_A \in L(N_j)$ and $l_B \in L(N_i)$:

$$\phi(l_A) > \phi(l_B)$$

DEFINITION 5. An I-chain is **consistent** when for any couple of nodes of the chain N_i, N_j, with $i, j \in (1,\ldots,k)$, $j > i$, the inequality (4) is valid.

The "group-ordering condition" implicit in the definition of consistency given above guarantees that the nodes of the I-chain bear on *disjoint* intervals of the response variable, to the benefit of the characterization of associated classes.

In an inconsistent I-chain I there always exists a set $R(I)$ of sets of leaves, which are called **residual sets** with the property that if we "ignore" all leaves belonging to any $r(i) \in R(I)$, then I appears to be consistent.

To a residual set $r(I) \in R(I)$ we assign a "penalty" $PEN(r(I))$ given by the number of sample cases attached to it, weighted on the basis of a *utility* of correctly classifying them. Given an I-chain, we may look for a (not-necessarily unique) *optimal* residual set $\hat{r}(I) \in R(I)$, i.e. the $r(I)$ with minimum penalty. The problem is then that of finding

$$\hat{I} = \min_{I \in I'} PEN(\hat{r}(I))$$

where I' denotes the set of I-chains on Θ with a certain restriction on the number of chain-nodes. The set of nodes in $\hat{I}$ will correspond to the final set of classes, and the logical descriptions for these classes will depend both on the structure of the tree and on the residual set.

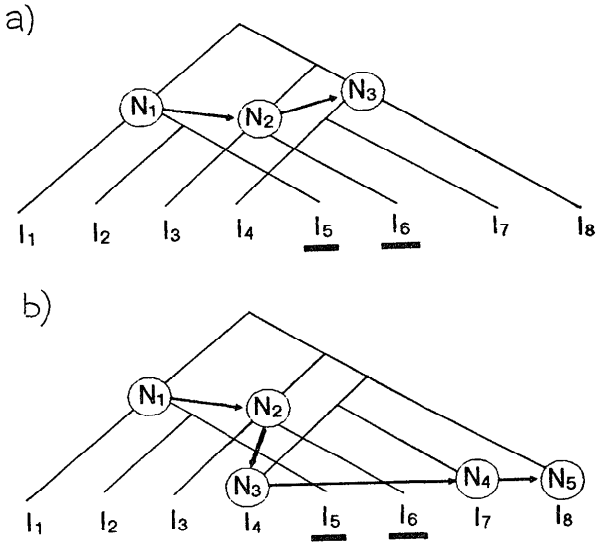


FIGURE 4

As an example, consider Figs. 4*a*,*b*, showing two alternative I-chains on the BRT obtained from MMM data. The one in Fig. 4b is obtained by iteratively expanding the one in Fig. 4a. The two I-chains share a common residual set. In fact, if we "ignore" l_5 and l_6, we find that the sets of leaves descending from the three nodes of the first I-chain, or from the five nodes of the second I-chain, lie on disjoint intervals of the response.

The following is an algorithm for finding a sub-optimal solution.

(1) select a set N of independent nodes of Θ
(2) define an I-chain I by ranking nodes in N according to the mean value of ϕ
(3) by means of algorithm FIND-RESIDUAL:
  (a) find an optimal residual set $\hat{r}(I)$
  (b) from I and $\hat{r}(I)$ derive a set of logical class descriptions
  set $PMIN = PEN(\hat{r}(I))$
(4) expand the I-chain by selecting a node and replacing it with its immediate successors, then reapply FIND-RESIDUAL. Proceed

iteratively with further expansions as long as there are expandable nodes in the chain and the minimum-penalty is not too high with respect to $PMIN$

(5) when a stable I-chain is reached, amalgamate classes which do not significantly differ in expected response.

8. ALGORITHM FIND-RESIDUAL

The crucial and computationally most difficult step of the algorithm described in the previous section is step (3). This step is managed through algorithm FIND-RESIDUAL, described in this section.

In order to make it suitable for object-oriented programming, this algorithm is based on a self-activated propagation mechanism, in which the nodes of the tree are viewed as autonomous processors, communicating locally via the links of the tree or of the I-chain.

To perform its autonomous computations, each node N_i of the I-chain uses a *working memory* containing: *(a)* two NL-dimensional arrays α_1 and β_1, *(b)* a scalar M_i, *(c)* a list L_i, *(d)* a list RES_i.

Each N_i is able to compute through a function *subs* the value of a logical variable $s_{ij} = subs(l_j, N_i)$, which is 1 (0) if l_j is (is not) a successor of N_i in Θ.

The algorithm has two phases. In the first, the computations are triggered by "messages" sent along the links of the I-chain. In the second phase, messages are sent along the edges of Θ.

Propagation along the I-chain

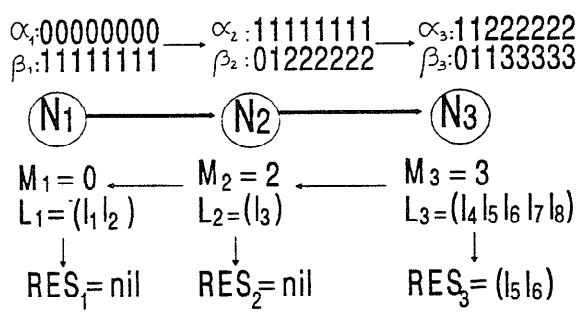


FIGURE 5

This process is illustrated by Fig. 5. Each N_i (except the root), upon receiving from N_{i-1} a message containing α_{i-1} and β_{i-1}, computes α_i and β_i iteratively:

$$\alpha_i(j) = \max\{\alpha_{i-1}(j), \beta_{i-1}(j)\}$$
$$\beta_i(j) = \max\{\alpha_i(j-1) + s_{ij}, \beta_i(j-1)\}$$

for $j = 1,...,NL$, assuming $\alpha_0(j) = \beta_0(j) = \alpha_i(0) = \beta_i(0) = 0$. Then N_i pairwise compares corresponding elements in α_i and β_i. Upon finding $\alpha_i(j) \geq \beta_i(j)$, it sets $M_i = j$. Finally, N_i sends α_i and β_i as a message to N_{i+1}.

The above propagation process is triggered by activating the root N_1 to compute:

$$\beta_1(0) = 0 \ , \ \alpha_1(j) = 0 \ ,$$
$$\beta_1(j) = \max\{s_{1j}, \beta_1(j-1)\}, \quad j = 1,...,NL$$

and to send to N_2 a message containing α_1 and β_1. After computing M_k, N_k sets $L_k = (l_{(M_k+1)}, \ldots, l_{NL})$ and triggers a "backwards" propagation by sending M_k as a message to N_{k-1}. The "backwards" propagation wave ripples along the I-chain by a simple mechanism: each N_i, upon receiving M_{i+1} from N_{i+1}, computes $L_i = (l_{(M_i+1)},...,l_{M_{i+1}})$, looks for all leaves $l_h \in L_i$ which are not among its own successors and puts them in the list RES_i. The final residual set is obtained by joining the RES-lists.

The algorithm above generates only one residual set, which may not be the optimal one. The extension of the algorithm to generate the full set of residual sets, which may be then searched for the optimal residual set, is straightforward, but its description is too lengthy to be included. It is available from the author upon request.

Propagation along the tree edges

This propagation generates logical descriptions for the classes represented by the nodes of the I-chain. Each node of the I-chain, say node N, interrogates its predecessor nodes in Θ to know the value assignments labeling edges on the path from N to the root. This conjunction of such assignments provides a logical description of the *general* class represented by N. Then it interrogates its own successors in Θ to know value assignments labeling the edges connecting N to the leaves falling in its own RES list. The conjunction of these latter value assignments yields a logical formula which tells how to discriminate from the general class represented by N those examples which fall in the residual set, i.e. that have response values which are more typical of other classes. For example, node N_1 has a general class description $(\neg A)$. Value assignments along the path from N_1 to the residual leave descending from it, l_5, are (H) and (C). The final class description will then be $(\neg A \ \neg(HC))$.

REFERENCES

[Aitkin *et.al.*, 1978] M.Aitkin, The Analysis of Unbalanced Cross-classifications, *J.R.Statist.Soc.A*, **141**, Part 2 (1978), 195-223.

[Breiman *et.al.*, 1984] L.Breiman *et.al.*, *Classification and Regression Trees*, Wadsworth International Group, Belmont, California (1984).

[Cox, 1972] D.R.Cox, Regression models and life tables, *J.Royal Stat.Soc.B*, **34** (1972), 187-208.

[Quinlan, 1983] J.R.Quinlan, Learning efficient classification procedures and their application to chess endgames, in: *Machine Learning: an AI approach* (R.S.Michalski, J.G.Carbonell, T.M.Mitchell eds.), Palo Alto, Calif.: Tioga (1983).

[Rendell, 1986] L.Rendell, Induction, of and by probability, in: *Uncertainty in Artificial Intelligence* (L.N Kanal, J.F.Lemmer eds.), Elsevier (1986).