

## Prediction trees with soft nodes for binary outcomes

Antonio Ciampi<sup>1,\*,\dagger</sup>, André Couturier<sup>1,2</sup> and Shaolin Li<sup>3</sup>

<sup>1</sup>*Department of Epidemiology & Biostatistics, McGill University, 1020 Pines Avenue West, Montreal H3A 1A2, HIT 1C8, Canada*

<sup>2</sup>*Montreal Heart Institute, Montreal (Quebec), Canada*

<sup>3</sup>*Department of Mathematics and Statistics, McGill University, Montreal (Quebec), Canada*

### SUMMARY

Consider the problem of predicting the occurrence of an event, the onset of diabetes mellitus, say, from a vector of continuous and discrete predictors. We propose a new algorithm for the construction of a tree-structured predictor for the event of interest, which uses a new approach for dealing with continuous predictors. The novelty is that the tree uses *splits* for continuous variables. This means that at each node an individual goes to the right branch with a certain probability, function of a predictor. The predictor as well as the particular shape of the function is chosen from the data by the proposed algorithm. We evaluate its performance on several real data sets, in particular comparing it with a standard tree-growing algorithm. We also present an analysis of a well-known data set, the Pima Indian diabetes data set, to illustrate the application of the method in biostatistics. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: predictive models; regression; neural nets; probabilistic split; latent classes; EM algorithm

### 1. INTRODUCTION

Methods for growing trees from data, also known as recursive partitioning, have become popular in many areas of application [1–3]. Their increasingly important role in the health sciences was recently reviewed in the insightful monograph by Zhang and Singer [4]. While early work emphasized single discrete or categorical outcomes, an important conceptual step towards broadening the scope of tree growing was to recognize that trees may be constructed for more complex outcomes [5], for example, censored survival times [5–7], count [8], longitudinal [9] and correlated multivariate data [10–12]. As a consequence, tree growing can now be seen as a family of techniques for adaptive data-driven prediction modelling, quite distinct in form but similar in aim to various forms of non-linear regression, for example,

---

\*Correspondence to: Antonio Ciampi, Department of Epidemiology and Biostatistics, McGill University, 1020 Pines Avenue West, Montreal H3A 1A2, HIT 1C8, Canada

†E-mail: antonio.ciampi@mcgill.ca

Contract/grant sponsor: Canadian National Science and Engineering Research Council

MARS [13] and neural networks [14]. We refer again to reference [4] for a comprehensive list of references to work on trees and other adaptive regression approaches in biostatistics.

The appeal of recursive partitioning is based on the considerable power with which a tree structure summarizes complex information in simple terms, and on the highly intuitive construction procedure. Suppose that the goal is to predict the probability  $p$  that an individual experience a specific event, for example, onset of diabetes mellitus, from a vector of predictors  $\mathbf{z}$ . The tree building algorithm starts by selecting a component  $z_k$  from  $\mathbf{z}$  and a yes/no question  $Q$  concerning  $z_k$ , then splits the data set according to whether the answer to  $Q$  is yes or no. The selection is based on some measure of information about  $p$ , so that the chosen split is the most informative one. The procedure is then repeated recursively, until a specified stopping rule terminates the process. An alternative to stopping rules is *pruning* [1]: a very large tree is constructed, a subsequence of this sub-tree is identified, and finally an element of this sequence is selected, termed *honest tree*. Both approaches share the same aim: reducing the *overfit bias*. Clearly, a recursive partitioning algorithm mimics a very common cognitive strategy whereby one sequentially acquires information by asking a series of questions, each question depending on the answer to the previous one, and each question locally maximizing the expected information about the goal.

Tree-growing algorithms are often claimed to emulate regression approaches in their ability to handle both continuous and discrete variables. However, the treatment of continuous variables remains somewhat unsatisfactory. First, the search of the optimal question for a continuous variable is reduced to the search of a cutpoint among all the observed values in the vector. Since such a search is much more extensive than the search for splits on discrete variables, continuous variables have an unfair advantage and as a result they tend to appear too often in the tree structure. Furthermore, repetitions of continuous variables in trees are a common occurrence. This may be due to an underlying relationship between a continuous variable and the outcome, other than the discontinuous one implied by the hard split (for example, linear, quadratic etc.). In this case, the algorithm attempts to imitate the relationship by repeating splits on the same variable. As a result, interpretation becomes more cumbersome, the effect of other important predictors may be masked and the resulting predictive power is lessened. The problem has been recognized by other authors. A substantial improvement, especially useful in biostatistics, has been proposed by Zhang and Singer [4] and implemented in their program RTREE: this consists in permitting the analyst to intervene in the automatic split-selection process so as to avoid or group together multiple splits on continuous variables. Secondly, as continuous variables often contain measurement errors, once the wrong decision is taken at a split, the error propagates along the tree structure and cannot be corrected. Partial remedies to this problem have been proposed; Quinlan [15] suggested probabilistic classification of new cases with uncertain data, and Ciampi *et al.* [16] developed a tree-growing algorithm to build a tree from data containing probabilistic imprecision, which however results in the construction of a standard tree, with probabilistic assignments of cases with imprecise data. Again, the flexibility of RTREE in tree construction may offer an important corrective in some situations. The problem, however, continues to puzzle some investigators and, although a universally convincing remedy is unlikely to be found, we believe that such questioning may help develop other useful approaches to tree growing.

In this paper we propose one such approach, inspired by the problem of handling continuous variables in tree growing: soft splits or, more precisely, probabilistic splits. The result is a new algorithm for the construction of a predictive model for a binary variable, with both

advantages and disadvantages as compared with standard trees. As we will see, it sacrifices some, but not all, of the intuitive appeal of an ordinary tree while achieving, in some cases, better predictions in terms of classification error, area under the ROC curve, deviance and Brier's score. Whether or not the unfair advantage of continuous variables is removed is hard to assess. However there is empirical evidence that this problem might be attenuated.

The proposed algorithm is adapted from the paper by Jordan and Jacobs [17], and modifies rather substantially the tree-growing algorithms. It can be seen as a compromise between traditional trees and approaches based on hybrid artificial neural net models [14]. We develop a modified RECPAM [10] algorithm and compare it to the standard tree algorithm CART on six well-known medical data sets, which are freely available from the Internet.

## 2. TREE WITH SOFT NODES

The following formulation of the problem of tree growing is valid both for standard trees with 'hard' nodes (hard trees), and for our new class of trees, that is, trees with 'soft' nodes (soft trees). Suppose we have a data matrix  $[Y|\mathbf{Z}]$ , where  $Y$  are observations of a binary outcome variable  $y$  (taking values 0/1, non-occurrence/occurrence of a specified event), and  $\mathbf{Z}$  of the predictors  $\mathbf{z} = (z_1, z_2, z_m) = (\text{age, gender, } \dots, \text{ systolic blood pressure})$ , on a random sample of subjects from a target population. The observation on the  $i$ th individual will be denoted  $(y_i, \mathbf{z}_i)$ : it is, obviously, the  $i$ th row of the data matrix. We will also define  $p(\mathbf{z}) = \Pr[y = 1|\mathbf{z}] = E[y|\mathbf{z}]$ . Assuming that  $p(\mathbf{z})$  can be adequately modelled by a tree structure, the problem is to estimate this structure from the data so that, given  $\mathbf{z}$ , we can predict the probability of event occurrence.

A hard tree structure for  $p(\mathbf{z})$  can be represented by a hierarchy of binary (1/0 or yes/no) questions concerning one predictor at a time. The questions are arranged in a diagram as the one shown in Figure 1. An individual with predictor vector  $\mathbf{z}$  is assigned to a chain of *nodes* (circles in the diagram), and, eventually, to one and only one *leaf* (the leaves are represented by the square boxes in the diagram), according to the answers provided to the questions. As a result, the predictor space is partitioned into subsets, represented by the leaves, such that the probability of outcome occurrence is homogeneous within each subgroup and can be represented by a leaf parameter,  $p_k$ .

In other words, an individual progresses from the *root node* (top circle) to his (unique) leaf, through a series of 'hard' decisions, of the type: 'go right' or 'go left'. Each decision is strictly determined by the answer to a question of the type 'is  $z_k$  in  $A$ ?' for a nominal predictor  $z_k$ , where  $A$  is a set of possible values or 'is  $z_j > a$ ?' for ordered predictors  $z_j$ , where  $a$  is a possible value.

A soft tree structure can be represented by a diagram very similar to that of Figure 1. The crucial difference is that, for ordinal variables, instead of making a 'hard' decision, of the type 'go left if  $z > a$  and go right otherwise', we make a 'soft' decision: 'go left with a certain probability; go right with the complementary probability'. It seems reasonable to require that the probability of going right increase towards 1, as  $(z - a)$  becomes positive, and that it decrease towards zero as  $(z - a)$  is negative and large in absolute value. As an example, consider the tree of Figure 2.

This simple tree has only two nodes and three leaves, to each of which the probability of an event is attached,  $p_k$ ,  $k = 1, 2, 3$ . Thus an individual of given age and sex is not assigned to a unique leaf but is distributed across the three leaves with probabilities defined by the

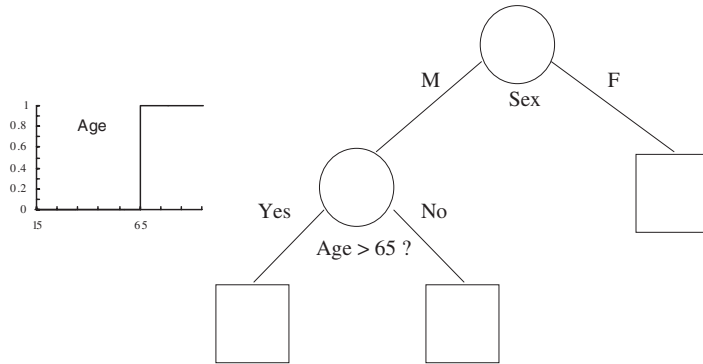


Figure 1. Hard tree structure. The graph to the left of the node associated to age represents the decision function of that node, with 1 corresponding to ‘go left’ and 0 to ‘go right’: it is a step function since the split is ‘hard’. The prediction equation associated to this tree is:  $p(\mathbf{z}) = p_1 I[\text{sex} = M] I[\text{age} > 65] + p_2 I[\text{sex} = M] I^c[\text{age} > 65] + p_3 I^c[\text{sex} = M]$ , where  $I[\dots]$  denotes a decision function and  $I^c = 1 - I$  its complement. Thus, for example,  $I[\text{age} > 65]$  is 1 for an individual over 65 and 0 for individuals less than 65.

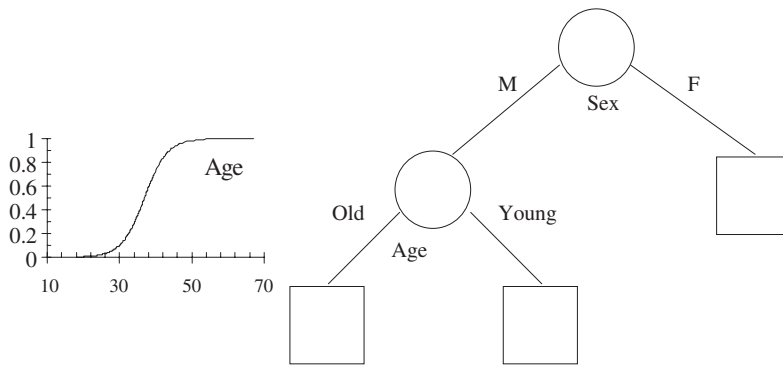


Figure 2. Soft tree structure. The graph corresponding to age represents now a soft threshold rather than a step function. While very old people and very young people have, respectively, probability (very nearly) 1 and 0 of ‘going left’, all those ‘in the middle’ can be thought of as being distributed right and left with the probability given by the decision function represented in the graph. The prediction equation associated to the tree is now  $p(\mathbf{z}) = p_1 I[\text{sex} = M] g(\text{age}) + p_2 I[\text{sex} = M] g^c(\text{age}) + p_3 I^c[\text{sex} = M]$  where now the function  $g$ , represented in the above graph, and its complement  $g^c = 1 - g$ , replace  $I$  and  $I^c$ , respectively.

tree structure; then  $p(\mathbf{z})$  is the weighted average of the  $p_k$ 's with the  $k$ th weight equal to the probability that the individual belongs to leaf  $k$ . This probability is easily computed from the tree diagram by recursively conditioning on the decision taken at each node of which the leaf is a descendant. See Figure 2 for the example and the Appendix for more details on the statistical model. Notice that to the ordinal variable age, the diagram does not associate a question. Instead, the name of the variable is written at the node and at the left and right

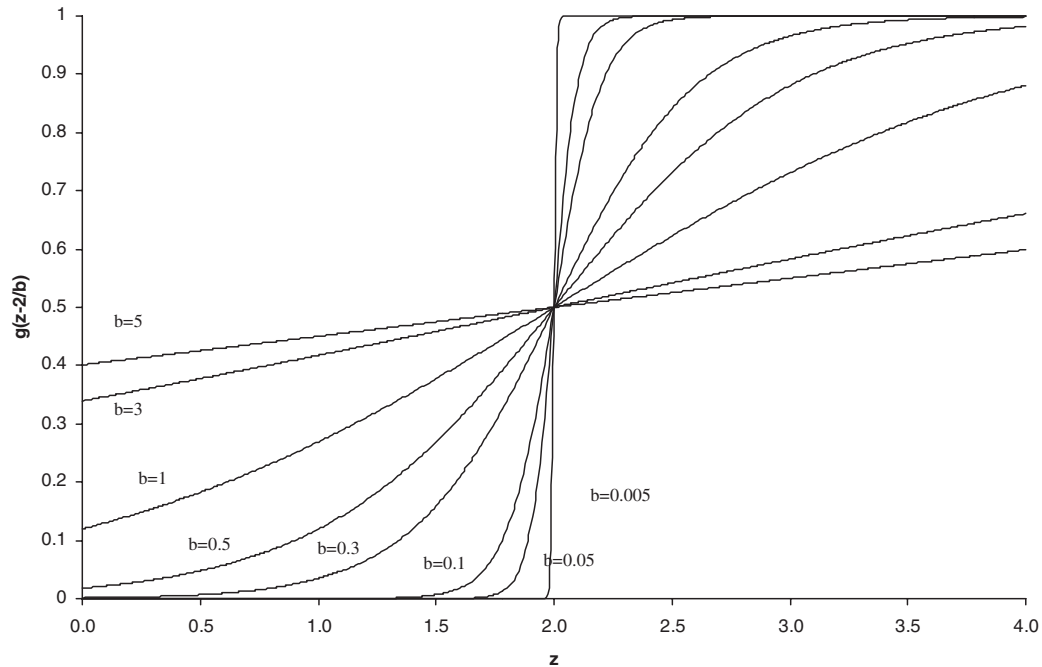


Figure 3. Logistic functions. The flexibility of the logistic function is apparent: it may mimic both a hard threshold, for very small values of the parameter  $b$ , and a linear function, for large  $b$ . The ‘interesting’ situation of a soft threshold is obtained for intermediate values of  $b$ .

branch issuing from it, the extreme values of the variable ‘old’ and ‘young’ are indicated, meaning that the extreme cases go to the left or to the right with high probability, while the in-between cases are distributed left or right with a certain probability, specified by the shape of the *sigmoid* function, called the *node decision function*. In this paper, we will use the *logistic* function  $g(z) = 1/(1 + \exp(-z))$  as our *node decision function*. Note that in a certain regions of the  $z$ -axis, the function is practically linear, and that by changing the scale of the  $z$ , the transition from 0 to 1 may be made as sharp as one wishes. It follows that if  $z$  only takes values in a finite interval, then one can chose the parameters of the *split decision function* so as to represent two extreme cases: a linear function and a step function. This is illustrated in Figure 3.

Two remarks are in order. First, the model proposed here is *not* a logistic regression model; instead, the probability of event occurrence ( $y = 1$ ) is modelled as a mixture of probabilities. However, ‘local’ logistic regressions (if  $g$  is chosen as in this paper) intervene in the definition of the mixing coefficients. Each logistic regression is based on a single variable selected by the tree construction algorithm. As remarked by a referee, it is natural to also think of constructing decision functions based on *all* the ordinal variables, a very useful suggestion for further work. One could also think of substituting the local logistic regressions with neural nets, as proposed in the machine learning literature (these are called *gating* networks in reference [17]). Furthermore, a dependence of the  $p_k$ ’s on the covariates may also be introduced (in the form of logistic regression or neural nets); this too can be achieved within

the general machine learning framework outlined in references [17, 18]. An approach of this type would most likely produce better predictions, but the resulting models become even more complex to interpret than the present one.

The second remark is aimed at showing that the soft tree proposed in this paper, though harder to understand than a standard 'hard' tree, may in some cases be justified on intuitive grounds. Consider for example the problem of predicting the probability of death in a given year on the basis of ordinal and continuous variables such as age and blood levels of several lipoproteins. Now, we may think of age as a proxy for a binary variable summarizing the (unknown) extent of coronary arterial damage (low/high) and we may conjecture that such a latent variable is much more directly related to the outcome in question than actual age. As for blood levels, we may argue that according to one trend in biomedical thinking, they may be regarded as proxy for the activation of certain genes. If we knew the extent of coronary arterial damage and the activation status for these genes, then we would have a set of binary variables from which to build a standard 'hard' tree, and such a model would have a certain intuitive plausibility. However, *given* a blood level of a lipoprotein, we may perhaps estimate the probability that the corresponding gene is activated; this corresponds to estimating a node decision function in a soft tree. Similarly, age may be the basis for inferring the extent of coronary arterial damage by modelling the appropriate node decision function. More generally, soft nodes may be convincing when there is reason to believe that there are latent classes underlying continuous predictors and that these latent classes are the important determinants of outcome. Clearly this may seem reasonable in some cases and far-fetched in many other situations, but this is typical of empirical model building. As all models, the soft tree would have to be checked carefully; the idea of enriching our approach by allowing all variables to participate in the decision at each node may be a useful way of checking the soft tree against a 'supermodel' which is more complex but still simpler than a 'black box' neural network.

In summary, our model is a compromise between a standard tree and the much richer, less interpretable, hybrid neural net model families which are becoming current in the machine learning literature; it can be seen as the first step in a step-up construction of a general hierarchy of experts' models.

### 3. A TREE-GROWING ALGORITHM FOR SOFT TREES

A standard tree-growing algorithm, such as CART or RECPAM, chooses automatically from the data the nodes, the questions defining the decision at each node, and the size of the tree. The algorithm presented in this section, for growing soft trees, is close in spirit to RECPAM, since it is model-based, see also Clark *et al.* [2]. The main idea is to recursively enlarge the current tree by adding one node so as to maximize the predictive ability of the enlarged tree, this is assessed on the basis of the underlying statistical model. The result is a sequence of nested trees with increasing apparent predictive ability. Among these, a specified criterion determines which one to choose in order to avoid overfitting while retaining a good predictive ability.

We will omit here the details about the numerical algorithm used for fitting a specified soft tree model. Interested readers are referred to the Appendix.

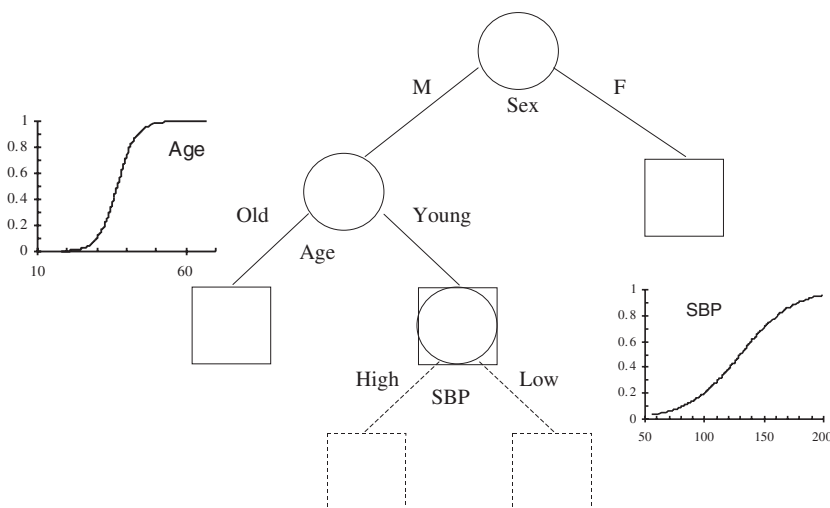


Figure 4. Basic operation for enlarging a tree. At each step the algorithm chooses the best node to split and the best splitting variable.

### 3.1. The tree-growing algorithm

The basic operation in the algorithm is to enlarge a given tree by adding to it one decision node. This is done by turning a leaf of the given tree into a node, which will be the ‘parent’ of two new leaves. This can be done in as many ways as there are leaves and as there are available splits at each leaf. Figure 4 shows graphically the basic operation for the tree of Figure 1, enlarged by adding a question concerning the variable systolic blood pressure (SBP).

In standard tree growing, an operation of this type is called *split*. In practice one wants to avoid splitting a leaf if its ‘children’ are too small and, in certain cases, if it has too few events of the outcome variable. A one-step augmentation will be called *admissible* if the expected number of subjects at the split node is at least  $s$ , for specified integer  $s$ . Notice that for standard trees this reduces to requiring a minimum number of subjects in the split node. Other admissibility conditions could also be imposed such as minimum number of events at a node, maximum number of missing values at a node etc.

From the pool of admissible splits at a specified node, we want to select the ‘best’ one. For this we need a measure of the information gained by increasing the complexity of the model. Let us define the information content of a tree  $T$  ( $IC(T)$ ) as the likelihood ratio statistic comparing the model associated to the tree, with the model associated to the trivial tree consisting of the root node only; see Kullback [19] for a general treatment of information in statistical modelling, and Ciampi [20] for an application to tree growing. A formal definition of the measure of information used is given in the Appendix.

We can now define a measure of the information gained ( $\Delta IC(T^*:T)$ ) by adding a split to the tree  $T$  as the difference in information content between  $T$  and the augmented tree  $T^*$ . The split with the highest  $\Delta IC$  will be selected as the ‘best’ split. A tree-growing algorithm follows:

1. FIX the admissibility condition  $s$ ; FIX the selection rule AIC or BIC.

2. STEP 0: INITIALIZE:  
Calculate the Information Content of the Trivial tree  $T_0$ .  
...
3. STEP  $K$ : ENLARGE THE TREE:  
Find the optimal admissible one-step augmentation  $T_k^*$
4. IF  $\Delta\text{IC}(T_k^* : T_k)$  is too small  
THEN STOP;  
ELSE, GO TO Step 3.

Two remarks are in order here. First, notice that unlike hard tree-growing, this soft tree-growing algorithm is *global*, in that, as soon as a soft node is introduced, all data points intervene in determining each subsequent step of the algorithm. This results in heavier calculations, but has the advantage of providing greater stability with respect to slight errors in the ordinal, as well as avoiding the counterintuitive notion of hard split on a continuous variable. For example, say the first split of a hard tree is age less than 65 and a 61-year-old patient is mistakenly coded as being 67. He would be classified with probability 100 per cent in the high-risk group. However, with a soft tree he might be classified in the high-risk group with, say, 60 per cent probability and in the low-risk group with 40 per cent probability, so that everything is not lost. Also, the implied difference between patients 64 and 66 years old is allowed to have a less dramatic impact.

Secondly, although not immediately apparent, the algorithm presented here does a kind of *pruning* which is done in parallel to the tree growing. This is, in fact, one of the pruning approaches described in the early RECPAM methodology paper [5]. We incorporated in the program two such methods to correct the overfitting bias, namely, the AIC (Akaike information criterion) [21] and the BIC (Bayesian information criterion) [22]. These are defined, in our case, as  $-2l_{\max} + \gamma^* n_l$ , where  $l_{\max}$  is the maximized log-likelihood,  $\gamma$  is the number of estimated parameters in the model and  $n_l$  is equal to 2 for the AIC and to the logarithm of the sample size for the BIC. Both the AIC and BIC curves have a minimum and often an elbow.

The tree-growing algorithm described earlier makes use, at choice, of any of these computationally light approaches to honest tree-selection. However it is not conceptually difficult to modify the algorithm, so as to make use of the split-sample or V-fold cross-validated information content curve to select the honest tree.

In summary, the present algorithm can be described as a hybrid of tree growing and adaptive step-wise regression; at each step, a *global* regression model is fitted (in tree growing the regression model is *local*) by *replacing* a term in the previous model (in adaptive step-wise regression one does not replace, but adds). The construction is *adaptive*, since, at each step, the term to be replaced is selected through a data-dependent process, and the replacing term is constructed from a variable, which is selected through, and transformed by, a data-dependent indicator or decision function.

## 4. COMPARISON OF SOFT AND HARD TREE APPROACHES

### 4.1. Cutpoint estimation on simulated data

We first compared the performance of the soft tree algorithm and the hard tree algorithm using two simulated experiments. In both experiments, 500 samples of size 1000 were generated

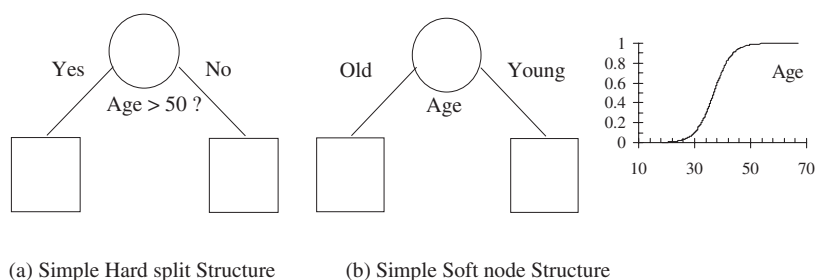


Figure 5. Simulated single split structures: (a) hard; (b) soft.

Table I. Cutpoint estimation using the hard and soft tree approach.

	Mean	Variance
<i>(a) Observations generated from a hard tree model with cutpoint fixed at 50</i>		
Hard tree	50.001	0.0429
Soft tree	49.988	0.1164
<i>(b) Observations generated from a soft tree model with fixed parameters <math>\alpha = -10</math>, <math>b = 0.2</math></i>		
Hard tree	50.24	6.2865
Soft tree	50.031	1.0041

from a simple tree structure with two leaves, based on a single continuous predictor  $z$ . In each case,  $Z$  was generated once and for all from a continuous random variable  $z$  uniformly distributed in  $[20, 80]$  (think of age).  $Y$  was generated anew for each sample, from a binary variable  $y$ , with  $p(z) = \Pr[y = 1|z]$  given by a tree model.

In the first experiment, data were simulated from a hard tree structure with hard split fixed at  $z = 50$  (see Figure 5(a)). In the second experiment, data were simulated from a soft tree structure as in Figure 5(b) with decision function  $g(-10 + 0.2z)$ , that is,  $g((z - a)/b)$ , with  $a = -\alpha/\beta = 50$ , and  $b = 1/\beta = 5$ . In each experiment we applied both the standard hard tree method, in which a cutpoint for the variable  $z$  is determined to define a split, and the soft tree approach, in which a decision function is determined by estimating the parameter  $\alpha$  and  $\beta$  in  $g(\alpha + \beta z)$ . The results are presented in Table I; mean and variance are estimated over the 500 trials.

These results indicate that the soft node approach performs well both when the model is correct and under model misspecification. In particular, when the data are generated by a hard tree model, the soft node estimator of the cutpoint  $(-\alpha/\beta)$  seems to have little bias, compared with the hard tree approach. As expected we observed a moderate increase in variance. In the second experiment, the hard tree approach has exactly the same behaviour under model misspecification. However, as shown from the variance estimate of 6.29, the hard tree structure has greater difficulty in estimating its cutpoint.

#### 4.2. Retrieving a simple structure

We have simulated 1000 samples of size 200 from the model shown in Figure 2. The model is a simple structure involving the 'artificial' demographic variables, age and sex. As with

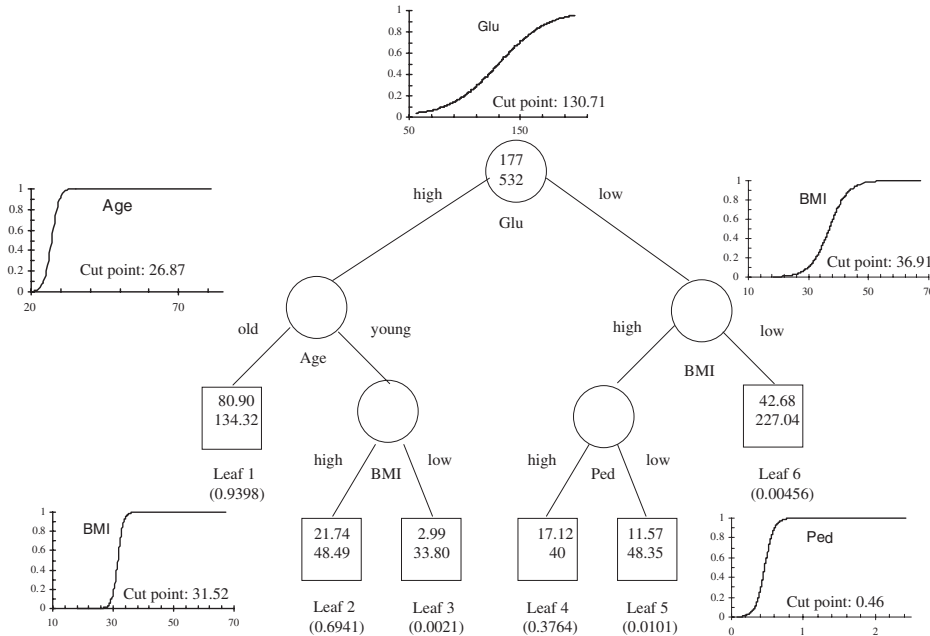


Figure 6. Soft tree structure for the Pima Indian data set.

our first simulation study, data were generated, with  $Z$  fixed and varying  $Y$  from sample to sample. The predictors were generated once and for all, with:  $\text{age} \sim \text{Uniform}[20, 80]$  and  $\text{Pr}[\text{sex} = \text{F}] = 0.3$  with decision function  $g(-10 + 0.2 \text{ age})$ . We expected that our new tree-growing algorithm would yield the right structure the majority of the time, that sometimes a smaller structure would be found, and only rarely either the null structure or a larger structure would be found.

For data generated from the model given in Figure 2, the soft tree algorithm found a two-leaf structure 31 per cent of the time, a three-leaf structure 66 per cent of the time, and a four-leaf structure only 3 per cent of the time. The parameters estimates averaged over the 1000 trials are:  $p_1 = 0.81$ ,  $p_2 = 0.72$ ,  $p_3 = 0.29$ . It is also interesting to note that when the hard tree-growing algorithm was applied to the simulated data, the averaged parameter estimates were:  $p_1 = 0.78$ ,  $p_2 = 0.72$ ,  $p_3 = 0.27$ ,  $-\alpha/\beta = 48.3$ . The averaged cutpoint estimate was  $-\alpha/\beta = 50.5$ .

Our experience shows that the two structures of Figure 6 are found in practically all cases when the sample size reaches 500.

### 4.3. Empirical study on real data

Six ‘real’ two classes data sets were used to compare the soft node approach to the hard node approach. All of them are available from the Internet either from the UCI Machine Learning Repository (Pima Indian, BUPA Liver Disorders, Heart Disease, Breast Cancer) at <http://www.ics.uci.edu/AI/ML/MLDBRepository.html>, or <http://hesweb1.med.virginia.edu/biostat/s/data/index.html> maintained by Dr Frank Harrell (Diabetes 2, Prostate Cancer). For the

Table II. Data set description.

	Original sample size	Effective sample size	Test set size	Number of predictors
Breast cancer	683	683	68	9 ( $C = 9$ ; $D = 0$ )
Pima Indian	532	532	50	7 ( $C = 7$ ; $D = 0$ )
Heart disease	303	296	30	8 ( $C = 5$ ; $D = 3$ )
Liver disease	345	345	35	5 ( $C = 5$ ; $D = 0$ )
Diabetes 2	403	367	37	8 ( $C = 6$ ; $D = 2$ )
Prostate cancer	502	482	48	13 ( $C = 10$ ; $D = 3$ )

$C$  = continuous predictor,  $D$  = discrete predictor.

Pima Indian data set we used Dr Ripley's version which can be found at <http://www.stats.ox.ac.uk/pub/PRNN>. Description of these data sets is also available from the web sites. Note that for the prostate cancer data set we used the patient status at 2 years, also the Ekg variable was not used. For all the other data sets the binary outcome was given. Table II gives a brief description of the data sets. All missing values for the selected predictors were removed since our program does not handle missing values yet. The effective sample size column in Table II shows the effect of removing the missing values. The test set sample size and the number of continuous and discrete predictors are also provided. All simulations were performed using S-plus 3.2 for Windows.

In accordance with Breiman's paper on Bagging [23], for each data set we repeated the following steps 100 times.

1. Randomly select 10 per cent of the data set and keep it apart as a Test set. The second part is the Practice set.
2. Use the Practice set to build two predictors (the soft node model and the hard node model).
3. Then, using the Test set, evaluate the model's performance by computing the test set misclassification error, the area under the ROC curve and the Brier score. We also looked at the tree size and deviance.

Model selection was performed using the AIC criterion for the soft tree and 10-fold cross-validation with the 0-SE rule for the hard tree. We first attempted using the AIC criterion for hard trees but we were putting the method at a disadvantage since it is well known that it tends to over-fit the model [15].

The results are shown in Tables III to VIII. Mean and standard deviation over the 100 trials are given, plus the minimum and maximum value obtained during the trials.

From Table III we see that except for the diabetes 2 data set the soft node predictor outperforms the hard node predictor. The improvement over the six data sets is on the average 4 per cent in favour of the soft tree approach. The most impressive result is obtained for the liver data set where we found a 13 per cent reduction in classification error. Using paired  $t$ -tests to compare the mean classification error on the same test set, we found the mean classification error of all the data sets other than diabetes 2 to be statistically different in favour of the soft node approach.

The hard tree method often produces pure nodes, that is, nodes for which the predicted probability of event is either 0 or 1. When predicting the probability of event for observations

Table III. Test set classification error.

	Soft tree				Hard tree				<i>p</i> -value*
	Mean	Std	Min	Max	Mean	Std	Min	Max	
Breast cancer	3.998	2.499	0	10.29	5.31	2.715	0	11.76	<0.0001
Pima Indian	22.86	5.58	12	34	26.12	5.797	10	38	<0.0001
Heart disease	25.37	7.429	3.33	43.33	33.73	7.803	10	56.67	<0.0001
Liver disease	37.74	7.556	20	54.29	50.63	8.634	22.86	71.43	<0.0001
Diabetes 2	15.68	5.34	2.7	27.03	14.62	5.47	2.7	27.03	0.0007
Prostate cancer	36.92	6.92	18.75	56.25	39.19	6.65	18.75	60.42	0.0013

\*Paired *t*-test for the difference in classification error.

Table IV. Test set deviance.

	Soft tree				Hard tree				<i>n</i> infinite*
	Mean	Std	Min	Max	Mean	Std	Min	Max	
Breast cancer	15.87	7.94	4.12	48.8	20.16	11.02	2.57	58.62	41
Pima Indian	48.03	9.11	32.63	71.08	50.22	8.05	29.66	68.66	12
Heart disease	33.08	7.93	17.39	64.24	38.25	7.01	21.63	55.13	7
Liver disease	45.52	4.09	36.33	58.84	49.33	2.98	40.61	63.85	1
Diabetes 2	28.51	8.08	12.17	57.25	29.56	7.83	14.42	56.16	5
Prostate cancer	66.01	7.78	49.72	85.73	64.16	4.38	55.63	80.36	1

\*The number of infinite test set deviance for the hard tree model.

Table V. Proportion of time the hard test set classification error ( $E_H$ ) is greater than the soft node test set classification error ( $E_S$ ).

	$E_H > E_S$	$E_H \geq E_S$
Breast cancer	58%	84%
Pima Indian	69%	77%
Heart disease	82%	86%
Liver disease	86%	93%
Diabetes 2	3%	78%
Prostate cancer	54%	66%

in the test set, sometimes a probability of 1 (or 0) is assigned to a subject who had no event (had an event). The consequence is an infinite test set deviance. Since this only happens for hard trees, we decided to show in Table IV the descriptive statistics for the finite test set deviance along with the number of time an infinite value occurred. From Table IV we conclude that the soft models performed well against the hard models.

Table V gives the proportion of time over the 100 repetitions that the hard tree test set classification error ( $E_H$ ) was greater than or greater than or equal to the soft node test set classification error ( $E_S$ ). Over the six data sets, the least amount of the time the soft node procedure is better or equal to the hard node procedure is 66 per cent. Thus the maximum number of times the soft approach was outperformed is 34 times out of the 100 trials (prostate

Table VI. Tree size.

	Soft tree				Hard tree			
	Mean	Std	Min	Max	Mean	Std	Min	Max
Breast cancer	5.01	0.86	3	8	8.76	3.66	4	17
Pima Indian	6.92	1.19	5	10	5.12	2.05	2	14
Heart disease	9.13	2.11	6	15	3.98	1.74	2	12
Liver disease	3.15	0.41	3	5	1.64	0.95	1	6
Diabetes 2	6	1.29	3	9	2.16	0.84	1	7
Prostate cancer	10.88	3.53	4	18	0.8	0.9	1	6

Table VII. Test set per cent area under the ROC curve (c-index).

	Soft tree				Hard tree			
	Mean	Std	Min	Max	Mean	Std	Min	Max
Breast cancer	99.13	0.82	94.74	100	97.23	1.96	91.76	100
Pima Indian	83.75	5.42	69.71	95.93	79.48	5.85	61.9	93.21
Heart disease	81.34	7.45	61.38	99.04	69.39	9.32	47.69	91.63
Liver disease	66.29	9.33	44.77	86.84	51.59	5.85	30.72	75.95
Diabetes 2	74.54	10.03	50	100	64.71	11.38	33.33	81.82
Prostate cancer	62.65	7.72	40.91	78.15	54.55	6.42	40.3	71.3

cancer). For diabetes 2 both methods yield the same classification error 76 per cent of the time. The soft approach was outperformed only 20 per cent of the time.

Table VI shows descriptive statistics for tree size. In general, the soft tree size is on the average larger than hard tree size but its variability is much smaller, which means that the soft approach yields more stable models.

The area under the ROC curve (Table VII) is another summary measure of the performance of a predictor. It corresponds to the probability of correctly identifying which of two patients randomly selected (one from the diseased group and one from the non-diseased group) came from the diseased group. The soft tree approach uniformly outperforms the hard tree approach, even for the diabetes 2 data set.

The Brier score (Table VIII) is the average of the squared difference between the prediction and the observed value. It is a measure of the performance of the model on predicting new cases. Once again the new method outperforms the hard tree method in achieving lower MSE.

## 5. ANALYSIS OF THE PIMA INDIAN DIABETES DATA SET

We will illustrate a soft tree analysis, using the Pima Indians diabetes data set modified by Dr Brian Ripley, which consists of 532 (out of 768 from the original data set) females at least 21 years old of Pima Indian heritage living near Phoenix, Arizona, U.S.A. The goal was to predict a binary variable which describes whether the patient shows signs of diabetes according to World Health Organization criteria (that is, if the 2-hour post-load plasma glucose

Table VIII. Test set Brier score (a.k.a. mean squared error).

	Soft tree				Hard tree			
	Mean	Std	Min	Max	Mean	Std	Min	Max
Breast cancer	0.0325	0.0165	0.005	0.086	0.0443	0.0197	0.0067	0.0956
Pima Indian	0.1572	0.0312	0.0985	0.2404	0.1714	0.0333	0.0937	0.2733
Heart disease	0.1778	0.03854	0.0785	0.2864	0.2227	0.0438	0.1168	0.3323
Liver disease	0.2292	0.0249	0.1692	0.2992	0.2552	0.0186	0.2134	0.3474
Diabetes 2	0.1199	0.0332	0.041	0.2075	0.122	0.0365	0.047	0.2271
Prostate cancer	0.2367	0.0288	0.1701	0.3296	0.2377	0.0215	0.1954	0.3149

Table IX. Simple summary statistics by disease categories.

Variable	Group	Mean $\pm$ Std	<i>p</i> -value
Number of pregnancies (npreg)	Diseased	4.701 $\pm$ 3.919	<0.0001
	Not diseased	2.927 $\pm$ 2.787	
Plasma glucose concentration (Glu)	Diseased	143.1 $\pm$ 31.27	<0.0001
	Not diseased	110 $\pm$ 24.29	
Diastolic blood pressure (bp)	Diseased	74.7 $\pm$ 12.52	<0.0001
	Not diseased	69.91 $\pm$ 11.9	
Triceps skin fold thickness (skin)	Diseased	32.98 $\pm$ 10.4	<0.0001
	Not diseased	27.29 $\pm$ 10.08	
Body mass index (BMI)	Diseased	35.82 $\pm$ 6.612	<0.0001
	Not diseased	31.43 $\pm$ 6.547	
Diabetes pedigree function (Ped)	Diseased	0.617 $\pm$ 0.399	<0.0001
	Not diseased	0.446 $\pm$ 0.299	
Age	Diseased	36.41 $\pm$ 10.84	<0.0001
	Not diseased	29.22 $\pm$ 9.903	

Methodology: Comparison of means is performed with one-way ANOVA.

was at least 200 mg/dl at any survey examination or if found during routine medical care). The National Institute of Diabetes and Digestive and Kidney Diseases provided the data. The following continuous predictors were available: number of pregnancies (npreg); plasma glucose concentration at 2 hours in an oral glucose tolerance test (Glu); diastolic blood pressure (mmHg) (bp); triceps skin fold thickness (mm)(skin); 2-hour serum insulin ( $\mu$  U/ml); body mass index (weight in kg/height in  $m^2$ ) (BMI); diabetes pedigree function (Ped); age (years). Serum insulin was removed from analysis because it had too many missing values.

In Table IX we provide summary statistics for the predictors of disease status. We can see that there is a strong association between all the predictors and the response variable.

The minimum AIC soft tree is shown in Figure 6 with the expected number of events and the expected number of subjects in each leaf along with the conditional probability of event

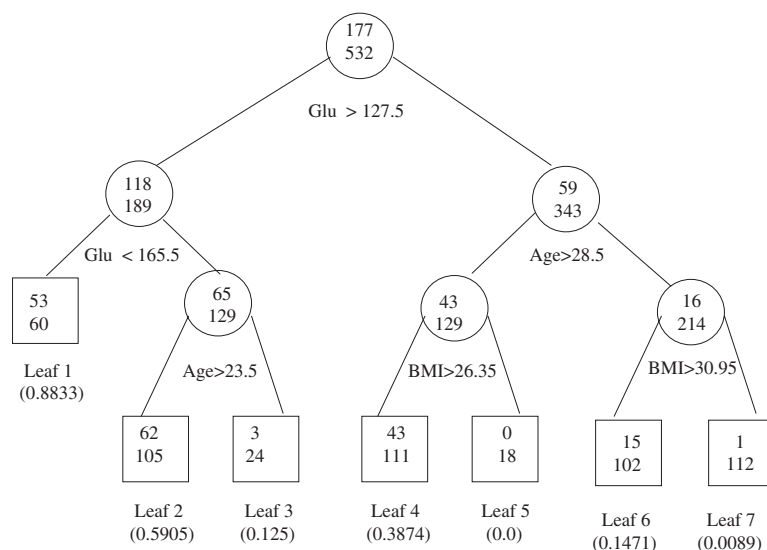


Figure 7. Hard tree structure for the Pima Indian data set.

in parentheses. It shows that Glu, Age, BMI and Ped play an important role in predicting disease, probably in a non-linear way, as indicated by the asymmetry of the tree (a highly symmetric tree would suggest that a linear model may be more appropriate to fit the data). Also, the presence of Ped at a low level may be seen as evidence of interactions, that is, Ped has predictive value only for a subgroup of patients. This model remains easy to interpret though slightly more complex than a hard tree (see Figure 7).

The plasma glucose concentration (Glu) shows a very soft split function, which means that a hard threshold might not be an appropriate representation of the relationship between Glu and diabetes. Patients with low plasma glucose concentration have high probability of being sent right. Those with plasma glucose concentration near the inflection point (130.71) are distributed to the left and right with probability given by the split function curve. For example, a patient with plasma glucose concentration of 150 has about 70 per cent chance of going left and 30 per cent chance of going right. Thus, by examining the split functions one can easily figure out the probability of ending into any of the leaf for a given patient, then compute the disease probability.

The hard tree of Figure 7 does not look very different from the soft tree. It contains the same variables, except Ped, and has an asymmetric structure. Glu also defines the first split, with a hard cut-off point of 127.5, not far from the inflection point of the soft split function of Figure 6. This difference is not large, nor surprising, in view of the fact, discussed in Section 4.1, that when a split is soft, the variability of the hard split is substantial. It should also be remarked that in this hard tree we observe that each of the three (continuous) variables is repeated twice, which suggests an attempt of the hard tree to mimic a soft one. In contrast, there is only one repeated variable in the soft tree (BMI).

Table X shows the re-substitution estimates of the four evaluation criteria. The order remains the same as the one found with the full validation approach (see Section 4.3), but,

Table X. Models' comparison.

	Soft tree	Hard tree
Deviance	453.67	448.23
Area under the ROC curve	86.54	85.81
Brier's score	0.14	0.14
Misclassification error	19.55%	21.05%

as it could be expected, the estimates are uniformly optimistic when compared to those in Tables III, IV, VII and VIII.

## 6. DISCUSSION

We have presented an approach to tree growing which replaces hard thresholds based on continuous variables with soft ones. As in standard trees, each node represents a decision, which sends a subject to the left or the right branch issuing from it. However, when the decision is based on a continuous variable, we propose, instead of a binary split, a probabilistic decision function sending individuals left or right with a probability depending on the value of the continuous variable. We have shown that, under reasonable restrictions on the shape of the decision function (sigmoid shape, well represented by the logistic function), it is possible to develop an algorithm that obtains the decision function from the data. Moreover, complete tree structures, with soft nodes when applicable, can be grown from the data. This can be done at a computational cost that, while higher than the cost of a standard tree construction, remains quite reasonable. On the other hand, some of the appeal of the standard tree is lost; although we feel that a soft node is not too hard to interpret, at least in certain cases (see Section 2), it is quite possible that to many other analysts the concept of soft node and the arguments given in this paper towards interpretation will appear contrived. Are the increased computational burden and the loss of interpretability offset by gains in predictive power? We showed on six data sets that the soft tree has a slight edge on the hard tree, while in some cases the hard tree performs better. We leave to the reader to decide whether this is enough to warrant further interest in the matter.

We feel that further effort to improve the method presented may be rewarding. For example, the fact that in some cases the hard tree yields a better predictor is puzzling: indeed the soft tree *contains* the hard tree as a limiting case (soft nodes tending to hard nodes), and therefore *in principle* one should always do better, or at least not worse, with the soft tree than with the hard tree. That this is not true, even in only a few cases, can only be a consequence of the fact that both hard and soft trees are obtained by algorithms that are not guaranteed to yield an optimal solution; thus, there is room for improvement.

However, even without further improvement, we feel that the idea of soft nodes and soft trees may be of some practical use in biostatistics. As a first general application, a soft node may replace the search for harsh cutpoints. Indeed, simulations indicate that the inflection point of the logistic function yields a reasonable, more stable estimate of a harsh cutpoint (see Section 4.1). This may not only speed up the computation, as an alternative to the updating formulae used in tree growing, but may help correcting the bias which favours continuous

variables over discrete ones in appearing in the tree structure. For example, the AIC associated with a soft tree penalizes more a continuous variable than the AIC associated with a standard tree as it counts the two additional parameters associated to the logistic function at each node. Also, since the search for the cutpoint is 'smooth' rather than discrete, the split statistic (information content) varies smoothly as a function of the parameter representing the inflection point, in contrast with the highly volatile behaviour of the split statistic in the discrete search.

While we have not given quantitative evidence that the soft tree solves the problems raised in the introduction, we clearly showed that our algorithm generally produces more efficient predictors than hard trees on a variety of data sets of clinical interest. This is true independently of the criterion of evaluation used (deviance, c-index, Brier's score and misclassification error); the most impressive improvement is obtained on the c-index (mean improvement of about 7.5 per cent). This gain in predictive power may well be due to a more efficient use of continuous variables (fewer splits on the same variable), as our example of Section 5 seems to suggest.

As another general application, soft trees may be useful in medical decision. A soft tree structure identifies just as incisively as a standard tree the key factors that should affect a decision, and gives as clear prescriptions for the 'extreme' cases. However, it also indicates that borderline cases should be treated with greater discretion and offers for them more cautious advice.

The method presented in this paper is at its initial stages of development. Further evaluative work is needed. This necessarily involves working with computer intensive approaches such as cross-validation and bootstrap. Thus statistical evaluation must be preceded by further acceleration of the basic EM algorithm used so far. Future work includes a generalization to multi-class problems and to other types of outcome such as continuous outcome and count data. Treatment of missing values also requires some generalization of the CART's approach.

## APPENDIX

### *A1. Statistical model*

To the graphical representation of a 'hard' tree, we can associate a statistical model, which will have the general form

$$p(\mathbf{z}) = p_1 I_1(\mathbf{z}) + p_2 I_2(\mathbf{z}) + \cdots + p_L I_L(\mathbf{z}) \quad (\text{A1})$$

where the  $I$ 's denote the indicator functions of the leaves of the tree:  $I_k(\mathbf{z}) = 1$  if  $\mathbf{z}$  belongs to the subset of the predictor space represented by the  $k$ th leaf. Clearly, this defines a mixture model with mixing coefficients depending (discontinuously) on the covariate vector  $\mathbf{z}$ . For the tree of Figure 1, it is easy to re-write equation (A1) as a products of binary variables representing the binary questions:

$$p(\mathbf{z}) = p_1 I[\text{sex} = \text{M}] I[\text{age} > 65] + p_2 I[\text{sex} = \text{M}] I^c[\text{age} > 65] + p_3 I^c[\text{sex} = \text{M}]$$

where we have used the following notation:  $I_A(z)$  is 1 if  $z$  is in  $A$  and 0 otherwise;  $A^c$  is the complement of  $A$ . Notice that  $I_A(z)$  represents the binary question 'is  $z$  in  $A$ ?'. This is true in general, although a precise notation for the general case would be extremely laborious and not needed in the present paper.

For soft trees, the equation of the statistical model for  $p(\mathbf{z})$  can be written down in a form similar to equation (A1):

$$p(\mathbf{z}) = p_1 J_1(\mathbf{z}) + p_2 J_2(\mathbf{z}) + \cdots + p_L J_L(\mathbf{z}) \quad (\text{A2})$$

where the  $J$ 's are no longer 'true' indicator functions. Instead, they are product of terms, one for each node, just as for the hard tree, except that, whenever the node is defined by an ordinal variable, the indicator function of the type  $I_A(z)$ , is replaced by a non-negative function with values in  $[0, 1]$ , which we shall call *split decision function*. The decision function has the form

$$g(z; a, b) = g\left(\frac{z - a}{b}\right) \quad (\text{A3})$$

where  $g$  is a *sigmoid* function, that is,  $g$  is non-decreasing with  $g(-\infty) = 0$  and  $g(\infty) = 1$ . For consistency, we also substitute to  $I_A^c(z)$  the complementary function  $g^c(z; a, b) = 1 - g(z; a, b)$ .

For the tree in Figure 2, equation (A2) would be

$$p(\mathbf{z}) = p_1 I[\text{sex} = \text{M}] g(\text{age}; a, b) + p_2 I[\text{sex} = \text{M}] g^c(\text{age}; a, b) + p_3 I^c[\text{sex} = \text{M}]$$

To summarize the general equation (A2), the  $J$ 's are products of sigmoid functions, for ordinal variables, and of indicator functions for nominal ones. The actual form of these product can be easily written out from the tree diagram. The  $J$ 's preserve the property of being non-negative and of summing to 1 for any value of  $\mathbf{z}$ . It follows that our model is a mixture model, with mixing coefficients which are smooth functions of the ordinal variables.

## A2. Fitting a soft tree model

To describe the algorithm in greater details, we have to discuss how a given tree model is fitted and how its predictive ability is assessed.

The problem to solve is the following. A soft tree structure is specified for a data matrix  $D = [Y|\mathbf{Z}]$ , in the form of equation (A2), except for the leaf probabilities  $p_1, p_2, \dots, p_L$ , and for the parameters  $(a_k, b_k)$ ,  $k = 1, \dots, n_c$ , appearing in the decision functions (A3) (one pair for each of the  $n_c$  nodes defined by an ordinal variable). We want to estimate these parameters by maximizing the log-likelihood of the data:

$$l(\theta) = \sum_{i=1}^N y_i \log p(\mathbf{z}_i) + (1 - y_i) \log q(\mathbf{z}_i) \quad (\text{A4})$$

where  $\theta$  denotes the vector of unknown parameters  $(a_k, b_k)$ ,  $p_1, p_2, \dots, p_L$ ,  $q = (1 - p)$ , and  $p(\mathbf{z})$  is given by (A2).

The explicit form of the likelihood is rather complex, because of the presence of the  $J$ 's in (A2). On the other hand, if the tree structure was hard rather than soft, with real indicator function  $I$ 's instead of  $J$ 's, the maximization of the likelihood would be trivial. One can think of the  $I$ 's as unobserved data. We can assume that for every soft node defined by an ordinal variable  $z_k$ , there is an unobserved binary variable  $\zeta_k$ , such that  $\zeta_k = 1$  represents the decision 'go left' and  $\zeta_k = 0$  the decision 'go right'. It follows that if we had 'complete data',  $(y_i, \mathbf{z}_i, \zeta_i)$ , we could associate a hard tree to the soft tree and estimate the  $p_k$  of equation (A1) rather easily. Seen this way, the natural method for solving this maximum likelihood estimation problem is the expectation-maximization (EM) algorithm [24], also used by Jordan and

Table A1. EM algorithm for soft tree estimation.

---

1. INITIALIZE:  
Initial values are assigned for the unknown parameters
2. E-STEP:  
Let  $\theta^{(r)}$  denote the current estimate of the parameters. Calculate  $E[\zeta_k|y; \theta^{(r)}] = \Pr[\zeta_k = 1|y; \theta^{(r)}]$  and substitute this value to  $\zeta_k$  in the complete likelihood.
3. M-STEP:  
Estimate the unknown parameter using maximum likelihood.
4. UPDATE  $\theta^{(r)}$
5. IF the difference between the updated and previous version of the parameter estimates is smaller than a fixed  $\varepsilon$ , STOP
6. ELSE, RETURN to step number 2, the E-step

---

Jacobs [17] in the context of neural networks. One possible drawback of using this technique instead of a more standard optimization method such as the Newton’s is its reputation of having slow convergence. However, as stated by Ripley [18], ‘It is unclear whether this reputation is justified as it may be much easier to find a nearly optimal solution (in the sense of high log-likelihood) than to find the maximizing parameters precisely’. For our simulations, this was not a major obstacle.

Rather than giving general formulae, we outline here the calculations for the tree of Figure 2. The model for the complete data can be written as

$$p(y = 1|\text{sex}, \text{age}, \zeta) = p_1^{\zeta I[\text{sex}=\text{M}]} p_2^{\zeta I[\text{sex}=\text{F}]} p_3^{(1-\zeta)} g((\text{age} - a)/b)^\zeta g^c((\text{age} - a)/b)^{1-\zeta}$$

giving the complete log-likelihood

$$\begin{aligned} l(\theta) = & \sum_{i=1}^n \{y_i \zeta_i I[\text{sex}_i = \text{M}] \log(p_1) + (1 - y_i) \zeta_i I[\text{sex}_i = \text{M}] \log(1 - p_1) \\ & + y_i (1 - \zeta_i) I[\text{sex}_i = \text{M}] \log(p_2) + (1 - y_i) (1 - \zeta_i) I[\text{sex}_i = \text{M}] \log(1 - p_2) \\ & + y_i I^c[\text{sex}_i = \text{M}] \log(p_3) + (1 - y_i) I^c[\text{sex}_i = \text{M}] \log(1 - p_3)\} \\ & + \sum_{i=1}^n \{\zeta_i \log(g((\text{age}_i - a)/b)) + (1 - \zeta_i) \log(g^c((\text{age}_i - a)/b))\} \end{aligned}$$

Notice that the first of the two terms in the above expression is simply a sum of three binomial log-likelihood’s, one for each leaf. The second term is easily recognized to be the log-likelihood for a logistic regression on the single variable age, provided we reparameterize the argument of the logistic function  $g$  to  $\alpha + \beta$  age, with  $\alpha = -a/b$  and  $\beta = 1/b$ . Table A1 gives a description of the algorithm which we used for our simulations.

During the E-step, we substitute to  $\zeta_i$  its expected value given  $y_i$  and  $z_i$ , and given the parameter vector evaluated at the current value of the parameter vector  $\theta^{(r)}$ . This is calculated from Bayes theorem as

$$\begin{aligned} \eta(y, \mathbf{z}, \theta^{(r)}) &= E[\zeta|y, \mathbf{z}, \theta^{(r)}] \\ &= \frac{g((\text{age} - a)/b) p_1^y (1 - p_1)^{1-y}}{g((\text{age} - a)/b) p_1^y (1 - p_1)^{1-y} + g^c((\text{age} - a)/b) p_2^y (1 - p_2)^{1-y}} \end{aligned}$$

The M-step consists in maximizing, with respect to  $\theta$

$$\begin{aligned} l(\theta) = & \sum_{i=1}^n \{y_i \eta_i(\theta^{(r)}) I[\text{sex}_i = \text{M}] \log(p_1) + (1 - y_i) \eta_i(\theta^{(r)}) I[\text{sex}_i = \text{M}] \log(1 - p_1) \\ & + y_i (1 - \eta_i(\theta^{(r)})) I[\text{sex}_i = \text{M}] \log(p_2) + (1 - y_i) (1 - \eta_i(\theta^{(r)})) I[\text{sex}_i = \text{M}] \log(1 - p_2) \\ & + y_i I^c[\text{sex}_i = \text{M}] \log(p_3) + (1 - y_i) I^c[\text{sex}_i = \text{M}] \log(1 - p_3)\} \\ & + \sum_{i=1}^n \{\eta_i(\theta^{(r)}) \log(g((\text{age}_i - a)/b)) + (1 - \eta_i(\theta^{(r)})) \log(g^c((\text{age}_i - a)/b))\} \end{aligned}$$

where we have set  $\eta_i(\theta^{(r)}) = \eta(y_i, \mathbf{z}_i; \theta^{(r)})$ . From the first term of the likelihood, we obtain the updated estimate for  $p_1$ :

$$p_1 = \frac{\sum_{i=1}^n y_i \eta_i I[\text{sex}_i = \text{M}]}{\sum_{i=1}^n \eta_i I[\text{sex}_i = \text{M}]}$$

and similar expressions for the other leaf probabilities. The second term is conveniently maximized by iterative reweighted least squares (or any of the well-known algorithms for logistic regression). This yields the updated estimates for  $\alpha$  and  $\beta$ .

In our experience, this approach yields reasonably stable estimates, independent of the initial condition, provided that we chose  $\varepsilon$  small enough, and provided that some care is taken in choosing the initial estimate for  $\alpha$  and  $\beta$ .

### A3. Measure of information

If  $T(\mathbf{D})$  denotes a tree constructed from the data matrix  $\mathbf{D}$ , we shall call the likelihood ratio statistic, the (*observed*) *information content* of  $T(\mathbf{D})$ ,  $\text{IC}(T(\mathbf{D}); \mathbf{D})$ . If we dispose of another data set  $\mathbf{D}'$ , we shall call the likelihood ratio statistics with parameter estimated on  $\mathbf{D}$  but calculated with the data of  $\mathbf{D}'$ , the *cross-validated information content* of  $T(\mathbf{D})$ , denoted  $\text{IC}(T(\mathbf{D}); \mathbf{D}')$ .

Suppose now that for any admissible one-step augmentation  $T'$  of a tree  $T$  we compute  $\text{IC}(T'(\mathbf{D}); \mathbf{D})$  and the information gain:  $\Delta\text{IC}(T' : T)$ . We will call *optimal (admissible) one-step augmentation* of  $T$ , the tree  $T^*$  such that  $\Delta\text{IC}(T^* : T) > \Delta\text{IC}(T' : T)$  for all (admissible) augmentations  $T' \neq T$ .

### ACKNOWLEDGEMENTS

Partially supported by the Canadian National Science and Engineering Research Council (NSERC).

### REFERENCES

1. Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and Regression Trees*. Wadsworth International Group: Belmont, California, 1984.
2. Clark LA, Pregibon D. Tree-based models. In *Statistical Models in S*, Chambers JM, Hastie T (eds). Wadsworth & Brooks: Pacific Grove, CA, 1992; 377–420.
3. Loh WY, Vanichsetakul N. Tree-structured classification via generalized discriminant analysis (with discussion). *Journal of the American Statistical Association* 1988; **83**(403):715–728.
4. Zhang H, Singer B. *Recursive Partitioning in the Health Sciences*. Springer-Verlag: New York, 1999.
5. Ciampi A, Chang CH, Hogg SA, McKinney S. Recursive partition: a versatile method for exploratory data analysis. In *Biostatistics*, McNeil IB, Umphrey GJ (eds). D. Reidel: New York, 1987; 23–50.

6. Davis R, Anderson J. Exponential survival trees. *Statistics in Medicine* 1989; **8**(8):947–962.
7. Segal MR. Regression trees for censored data. *Biometrics* 1988; **44**(1):35–48.
8. Ciampi A, Lou Z, Lin Q, Negassa A. Recursive partition and amalgamation with the exponential family: theory and applications. *Applied Stochastic Models and Data Analysis* 1991; **7**(3):121–137.
9. Segal MR. Tree-structured methods for longitudinal data. *Journal of the American Statistical Association* 1992; **87**(418):407–419.
10. Ciampi A. Constructing prediction trees from data: the RECPAM approach. *Proceedings from the Prague 1991 Summer School on Computational aspects of model choice*, Physica-Verlag, Heidelberg, 1992, 105–152.
11. Ahn H, Chen JJ. Tree-structured logistic model for over-dispersed binomial data with application to modeling developmental effects. *Biometrics* 1997; **53**(2):435–455.
12. Zhang HP. Classification trees for multiple binary responses. *Journal of the American Statistical Association* 1998; **93**(441):180–193.
13. Friedman JH. Multivariate adaptive regression splines. *Annals of Statistics* 1991; **19**(1):1–141.
14. Bishop C. *Neural Network for Pattern Recognition*. Clarendon Press: Oxford, 1995.
15. Quinlan JR. Probabilistic decision trees. In *Machine Learning III*, Kodratoff Y, Michalski R (eds). Morgan Kaufman Publisher: San Francisco, 1990; 140–152.
16. Ciampi A, Diday E, Lebbe J, Périnel E, Vignes R. Tree-growing with probabilistically imprecise data. In *Ordinal and Symbolic Data Analysis*, Diday E, Lechevallier Y, Opitz O (eds). Springer: New York, 1996; 201–212.
17. Jordan MI, Jacobs RA. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation* 1994; **6**(2):181–214.
18. Ripley BD. *Pattern Recognition and Neural Networks*. Cambridge University Press: New York, 1996.
19. Kullback S. *Information Theory and Statistics*. Wiley: New York, 1968.
20. Ciampi A. Generalized regression trees. *Computational Statistics and Data Analysis* 1991; **12**(1):57–78.
21. Akaike H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 1974; **AC-19**:716–723.
22. Schwartz G. Estimating the dimension of a model. *Annals of Statistics* 1978; **6**(2):461–464.
23. Breiman L. Bagging predictors. *Machine Learning* 1997; **26**(2):123–140.
24. Dempster AP, Laird NM, Rubin DB. Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B* 1977; **39**(3):311–319.