LOTUS: An algorithm for building accurate and comprehensible logistic regression trees

Journal of Computational and Graphical Statistics, 2004, **13**, 826-852

Kin-Yee Chan and Wei-Yin Loh

Logistic regression is a powerful technique for fitting models to data with a binary response variable, but the models are difficult to interpret if collinearity, nonlinearity, or interactions are present. Besides, it is hard to judge model adequacy since there are few diagnostics for choosing variable transformations and no true goodness-of-fit test. To overcome these problems, we propose to fit a piecewise (multiple or simple) linear logistic regression model by recursively partitioning the data and fitting a different logistic regression in each partition. This allows nonlinear features of the data to be modeled without requiring variable transformations. The binary tree that results from the partitioning process is pruned to minimize a cross-validation estimate of the predicted deviance. This obviates the need for a formal goodness-of-fit test. The resulting model is especially easy to interpret if a simple linear logistic regression is fitted to each partition, because the tree structure and the set of graphs of the fitted functions in the partitions comprise a complete visual description of the model. Trend-adjusted chi-square tests are used to control bias in variable selection at the intermediate nodes. This protects the integrity of inferences drawn from the tree structure. The method is compared with standard stepwise logistic regression on thirty real datasets, with several containing tens to hundreds of thousands of observations. Averaged across the datasets, the results show that the method reduces predicted mean deviance by nine to sixteen percent. We use an example from the Dutch insurance industry to demonstrate how the method can identify and produce an intelligible profile of prospective customers.

Key Words: Piecewise linear logistic regression; Recursive partitioning; Trend-adjusted chi-square test; Unbiased variable selection.

Kin-Yee Chan is Assistant Professor, Department of Statistics and Applied Probability, National University of Singapore, Block S16 Level 7, 6 Science Drive 2, Singapore 117546 (Email: kinyee@stat.nus.edu.sg).Wei-Yin Loh is Professor, Department of Statistics, University of Wisconsin-Madison, 1210 West Dayton Street, Madison, Wisconsin 53706 (Email: loh@stat.wisc.edu).

1 INTRODUCTION

Logistic regression is a well-known statistical technique for modeling binary response data. In a binary regression setting, we have a sample of observations on a 0-1 valued response variable Y and a vector of K predictor variables $\mathbf{X} = (X_1, \ldots, X_K)$. The linear logistic regression model relates the "success" probability $p = P(Y = 1)$ to $\mathbf{X}$ via a linear predictor $\eta = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_K X_K$ and the logit link function $\eta = \text{logit}(p) = \log\{p/(1-p)\}$. The unknown regression parameters β_0, β_1, ..., β_K are usually estimated by maximum likelihood. Although the model can provide accurate estimates of p, it has two serious weaknesses: (1) it is hard to determine when a satisfactory model is found, because there are few diagnostic procedures to guide the selection of variable transformations and no true lack-of-fit test, and (2) it is difficult to interpret the coefficients of the fitted model, except in very simple situations.

A good example of the difficulties is provided by the low birth weight dataset of Hosmer and Lemeshow (1989, Appendix 1). The data are from a study to identify the risk factors associated with babies born with low birth weight (defined as less than 2500 grams). There are 189 women in the data, with 59 giving birth to babies with low birth weight. The variables are listed in Table 1 with `low` being the Y variable.

Hosmer and Lemeshow (1989) and Venables and Ripley (1999) use transformations and stepwise variable selection techniques to fit logistic regression models to these data. Their models are displayed in Table 2. Both find the transformed variable $\mathtt{ptd} = I(\mathtt{ptl} > 0)$ to be highly significant, but they differ in other respects. For example, `race` is included in one but not the other, and similarly for `ftv`. The `smoke` variable appears in two places in both models: as a main effect term and an interaction term. Although the main effect of `smoke` is positive, it is dominated by a negative interaction term in both models. This leads to the implausible conclusion that smoking reduces the probability of low birth weight in some circumstances! The reasons for these difficulties in interpretation are well-known. They are nonlinearity, collinearity, and interactions among the variables and bias in the coefficients due to selective fitting. The latter makes it risky to judge the significance of a variable by its t-statistic.

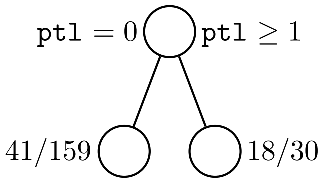

Figure 1: A logistic regression tree with a multiple linear logistic model in `age` and `lwt` in each of its two leaf nodes. The fraction beside each leaf node is the number of low birth weight cases divided by the node sample size.

Table 1: Variables for birth weight data

Name	Values
low	1 if baby has low birth weight, 0 otherwise
age	age of mother in years
lwt	weight of mother (lbs) at last menstrual period
race	1 if white, 2 if black, 3 if other
smoke	1 if smoking during pregnancy, 0 otherwise
ptl	number of previous premature labors
ht	1 if there is history of hypertension, 0 otherwise
ui	1 if presence of uterine irritability, 0 otherwise
ftv	number of physician visits in first trimester

One way to avoid these problems without sacrificing estimation accuracy is to partition the sample space and fit a linear logistic regression model containing only one or two untransformed variables in each partition. We call this a logistic regression tree model. Figure 1 shows the result for the present data. There are just two partitions, defined by whether or not ptl is zero (note that this coincides with the definition of the human-transformed variable ptd). One hundred fifty-nine women have ptl value zero and thirty have values greater than zero. A logistic regression model linear in age and lwt is fitted to each partition.

Figure 2 shows the data points and contour plots of the estimated probability $\hat{p}$. Clearly $\hat{p}$ decreases with increase in age and lwt. Women with positive ptl values are two to three times as likely to give birth to babies with low birth weight. These women tend to be younger and have lower values of lwt. The different heights and orientations of the contour lines indicate vividly that there is an interaction among the three variables. The absence of the other predictors in the model suggests that their importance is secondary, after ptl, age, and lwt are accounted for. Thus the model not only provides a prediction formula for $\hat{p}$, but it also conveys visually interpretable information about the roles of the predictor variables. Note that the model complexity is being shared between the tree structure and the regression models in the nodes. This division of labor helps to keep the logistic regression tree model simple.

Algorithms for logistic regression trees (also known as hybrid trees or model trees in the machine learning literature) evolved from attempts to obtain predictions of class-membership probabilities from classification trees. A naive approach would use the sample proportions at the leaf nodes of the trees but this has two major disadvantages. First, because the estimate of p is piecewise-constant, it will not be accurate unless the tree is large. But a large tree is harder to interpret than a small one. This problem is worse when the class proportions are highly unequal, for the tree may have no splits after pruning.

Table 2: Two logistic regression models fitted to birth weight data. The derived variables are: $\texttt{ptd} = I(\texttt{ptl} > 0)$, $\texttt{lwd} = I(\texttt{lwt} < 110)$, $\texttt{race1} = I(\texttt{race} = 2)$, $\texttt{race2} = I(\texttt{race} = 3)$, $\texttt{ftv1} = I(\texttt{ftv} = 1)$, and $\texttt{ftv2+} = I(\texttt{ftv} \geq 2)$.

Hosmer and Lemeshow (1989, p. 101)				Venables and Ripley (1999, p. 224)			
Variable	Coeff.	S.E.	t-value	Variable	Coeff.	S.E.	t-value
Intercept	-0.512	1.088	-0.47	Intercept	-0.583	1.419	-0.41
`age`	-0.084	0.046	-1.84	`age`	0.076	0.054	1.40
`lwd`	-1.730	1.868	-0.93	`lwt`	-0.020	0.007	-2.73
`ptd`	1.232	0.471	2.61	`ptd`	1.560	0.500	3.15
`smoke`	1.153	0.458	2.52	`smoke`	0.780	0.419	1.86
`ht`	1.359	0.662	2.05	`ht`	2.066	0.747	2.76
`ui`	0.728	0.480	1.52	`ui`	1.818	0.665	2.73
`race1`	1.083	0.519	2.09	`ftv1`	2.921	2.279	1.28
`race2`	0.760	0.460	1.63	`ftv2+`	9.242	2.632	3.51
`age`×`lwd`	0.147	0.083	1.78	`age`×`ftv1`	-0.162	0.096	-1.68
`smoke`×`lwd`	-1.407	0.819	-1.72	`age`×`ftv2+`	-0.411	0.118	-3.50
				`smoke`×`ui`	-1.916	0.971	-1.97

Second, linear trends in p are notoriously difficult for a piecewise-constant tree to model (Steinberg and Cardell 1998).

Quinlan's (1992) M5 is a regression tree algorithm designed for a continuous-valued Y variable. An M5 tree is basically a classification tree with linear regression functions at the leaf nodes. First, an ordinary classification tree is constructed, with the standard deviation of the Y values used as node impurity function. Then the tree is pruned, with a stepwise linear regression model fitted to each node at every stage. Although this method is applicable to data where the values of Y are 0 or 1, it does not guarantee that the estimated probabilities lie within these limits. The obvious solution of substituting logistic regression for linear regression has not been attempted. A possible difficulty is the conflicting aims of classification and logistic regression—classification trees prefer splits that cleanly separate the classes but such splits yield datasets that cannot be fitted by logistic regression.

Steinberg and Cardell (1998) proposed the hybrid CART-Logit model to improve the prediction accuracy of logistic regression by combining its strengths with that of CART (Breiman, Friedman, Olshen and Stone 1984). First they build a CART tree model. Then the original predictor variables and information from the CART tree, such as leaf node identifiers and predicted probabilities, are used as inputs to a stepwise logistic regression model. While it is unclear whether this approach produces more accurate probability estimates, it is clear that the model is much more complex and hence harder to interpret than a standard logistic regression model.

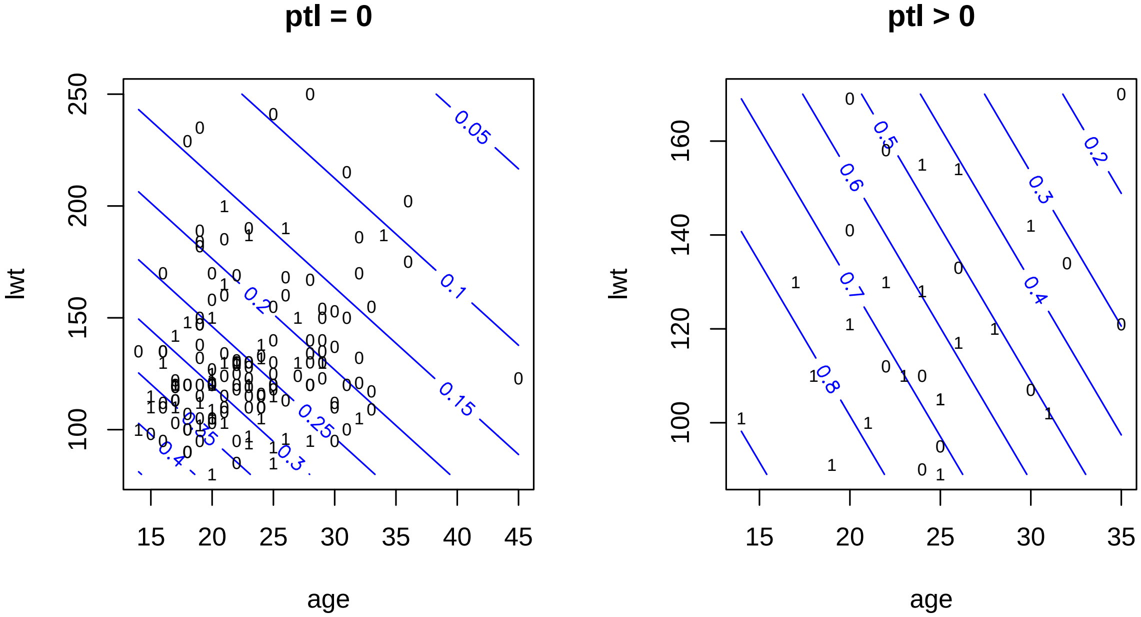

Figure 2: Data and contour plots of $\hat{p}$ in the leaf nodes of the tree in Figure 1.

Both M5 and CART-Logit require the building of a classification tree as a preliminary step. A second school of thought aims at recursively partitioning the data and fitting logistic regression models to the partitions formed at each step. Although this is a more natural approach, very few such algorithms have appeared in the literature. The main reason is that the usual exhaustive search technique (as practiced in CART and M5) is very compute-intensive for logistic models—two logistic models must be fitted to the partitions induced by *each* candidate split. Chaudhuri, Lo, Loh and Yang (1995) solve this problem by adapting a residual-based approach proposed by Chaudhuri, Huang, Loh and Yao (1994) for least squares regression trees. The latter selects the variable for which the signs of the residuals appear most non-random, as determined by the significance probabilities of two-sample t tests. In binary regression, however, the signs of the residuals, $Y - \hat{p}$, do not change with the fitted model. Chaudhuri et al. (1995) overcome this difficulty by using the signs of “pseudo-residuals”, defined as $\tilde{p} - \hat{p}$, where $\tilde{p}$ is a smoothed nearest-neighbor estimate of p. Although this solves the problem, it creates two new problems of its own: (1) the method is sensitive to the degree of smoothing and (2) the smoothing requirement renders it inapplicable to datasets with categorical (i.e., unordered) variables.

In the opposite direction, linear logistic regression has also been used as a classifier to predict Y instead of estimating p. For this purpose, Lim, Loh and Shih (2000) find that its classification accuracy is excellent compared to other classification methods on many real datasets. Perlich, Provost and Simonoff (2003) show, however, that there is a cross-over effect between logistic regression and C4.5, a classification tree algorithm. The classification accuracy of logistic regression is very good for small to moderate sized datasets but it levels off as the sample size increases. On the other hand, the accuracy of C4.5 begins lower but keeps increasing with sample size. The reason for the difference in performance may be due to the number of variables in a logistic regression model being fixed whereas the number of nodes in a C4.5 model grows with the sample size.

In this article, we present a new method called LOTUS (for *Lo*gistic *T*ree with *U*nbiased *S*election) for the automatic construction of logistic regression trees. By allowing the option of fitting only simple linear logistic regressions in the nodes, the tree model is visualizable and hence more comprehensible than standard multiple linear logistic regression. Further, because the number of parameters in a LOTUS model increases with sample size, it is not expected to have the leveling effect observed by Perlich et al. (2003). LOTUS fits a linear logistic model where the latter is best—in each node, where the sample size is never large. In addition, LOTUS has five properties that make it desirable for analysis and interpretation of large datasets: (1) negligible bias in split variable selection, (2) relatively fast training speed, (3) applicability to quantitative and categorical variables, (4) choice of multiple or simple linear logistic node models, and (5) suitability for datasets with missing values.

The remainder of the paper is organized as follows. Section 2 describes the roles a predictor variable can take in LOTUS and the types of logistic regression models that can be fitted at the nodes. Section 3 discusses the selection bias problem and details our solution. Section 4 presents simulation results to demonstrate its effectiveness under various conditions. Section 5 considers split point selection and Section 6 deals with missing values and pruning. Section 7 compares the predictive accuracy and training time of LOTUS with that of stepwise logistic regression on thirty real datasets, some with hundreds of thousands of observations. Averaged across the datasets, LOTUS reduces the predicted mean deviance of stepwise logistic regression by nine to sixteen percent. LOTUS is sometimes faster but no more than thirty times slower on these datasets. Section 8 shows how LOTUS is used to identify and create an intelligible profile of caravan policy holders in a dataset from the Dutch insurance industry. Finally, Section 9 concludes the article with some remarks.

2 PREDICTOR ROLES AND NODE MODELS

LOTUS can use categorical as well as quantitative variables. Categorical variables may be ordinal (called `o`-variables) or nominal (called `c`-variables). The traditional method of dealing with nominal variables is to convert them to vectors of indicator variables and then use the latter as predictors in a logistic regression model. Since this can greatly increase the number of parameters in the node models, LOTUS only allows categorical variables to participate in split selection; they are not used as regressors in the logistic regression models.

LOTUS allows the user to choose one of three roles for each quantitative predictor variable. The variable can be restricted to act as a regressor in the *fitting* of the logistic models (called an `f`-variable), or be restricted to compete for *split* selection (called an `s`-variable), or be allowed to serve both functions (called an `n`-variable). Thus an `n`-variable can participate in split selection during tree construction and serve as a regressor in the logistic node models. For example, we ensured that each node model is relatively simple in Figure 1 by setting `age` and `lwt` as `n`-variables, `race`, `smoke`, `ht`, and `ui` as `c`-variables, and `ptl` and `ftv` as `s`-variables.

LOTUS can fit a multiple linear logistic regression model to every node or a best simple linear regression model to every node. In the first option, which we call `LOTUS(M)`, all `f` and `n`-variables are used as linear predictors. In the second, which we call `LOTUS(S)`, each model contains only one linear predictor—the one among the `f` and `n`-variables that yields the smallest model deviance per degree of freedom. *Deviance* is a standard measure of variation in the literature of generalized linear models; see, e.g., McCullagh and Nelder (1989). It is also the impurity measure used in the S-PLUS `tree` function (Clark and Pregibon 1992). Suppose that ℓ_M and ℓ_S denote the maximized log-likelihood values for the model of interest and the saturated model (i.e., the most complex model having as many parameters as observations), respectively. Then the deviance of a generalized linear model is defined as $D = -2(\ell_M - \ell_S)$. For logistic regression, the deviance simplifies to

$$D = -2\sum_{i=1}^{n} [y_i \log(\hat{p}_i/y_i) + (1-y_i)\log\{(1-\hat{p}_i)/(1-y_i)\}] \qquad (1)$$

where $\hat{p}_i$ is the estimated probability for the ith observation. This function behaves like the residual sum of squares for least squares regression models; it is non-negative and decreases as more variables are added to the model. The total impurity for a tree is obtained by summing the deviances in all the partitions. *Degrees of freedom* is defined as the number of fitted observations minus the number of estimated parameters, including the intercept term. The tree in Figure 1 is obtained with the `LOTUS(M)` option. The algorithm for `LOTUS(S)` may be stated as follows.

Algorithm 1 Best simple linear logistic model option (`LOTUS(S)`)
Suppose $X_1, \ldots, X_K$ are the `f`- or `n`-variables. The following steps are carried out at each node.

1. *For each $k = 1, \ldots, K$, fit the model $\log\{p/(1-p)\} = \beta_0 + \beta_1 X_k$. Let D_k denote its deviance as defined in (1), and let ν_k denote its degrees of freedom. Compute $\bar{D}_k = D_k/\nu_k$. If the data are pure with respect to the Y values or if the model does not converge, define $\bar{D}_k = \infty$.*

2. *Let k^* be the smallest k that minimizes $\bar{D}_k$.*

(a) If $\bar{D}_{k^*} = \infty$, delete the node and its sibling and turn its parent into a leaf node.

(b) Otherwise, select the simple linear logistic model with predictor X_{k^*}.

`LOTUS(S)` has the advantage of permitting the estimated logistic function in each node to be visualized via a plot against the selected predictor (see Section 8 for an example). It is computationally faster than `LOTUS(M)` when there are very large numbers of observations or large numbers of predictor variables. Further, because only one predictor variable is employed in each node, `LOTUS(S)` is much less likely to encounter problems with complete or quasi-complete separation of the data (Allison 1999, p. 41) than `LOTUS(M)`.

3 UNBIASED SPLIT VARIABLE SELECTION

It is critically important for a classification or regression tree method to be free of selection bias. If it is not, grossly misleading conclusions about the effects of predictor variables may result. We first give an example of selection bias in the CART algorithm and then show how the problem is controlled in LOTUS. Our example employs the `bands` dataset from the UCI data repository (Blake and Merz 2000). The problem is to predict cylinder banding during rotogravure printing. The original dataset consists of observations on 40 variables from 512 cylinders. Half of the variables are quantitative and half categorical. The response variable is band type, which takes values `band` ($Y = 0$) and `noband` ($Y = 1$). To avoid confounding selection bias effects with missing value effects, we restrict ourselves here to the subset of 277 cases with complete values. We also drop two variables, `timestamp` and `cylinderID` because they are unique or almost unique case identifiers. This leaves 19 quantitative and 18 categorical predictor variables. Two of the latter take many values, namely, `jobno` and `customer` with 148 and 51 values, respectively.

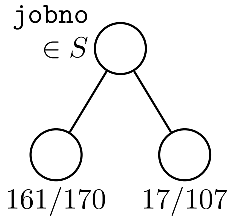

Figure 3: CART tree for the bands data. The set S contains 89 out of 148 job numbers. At each decision node, a case goes into the left branch if and only if the stated condition is true. The predicted value of $P(Y = 1)$ is given beneath each leaf node as the number of cases with $Y = 1$ divided by the node sample size. The cross-validation estimate of misclassification cost is 0.296.

Figure 3 shows the CART 1-SE tree constructed with the 37 predictor variables. It splits only once, on `jobno` which has $2^{147} - 1 \approx 10^{44}$ allowable splits. If we remove `jobno`,

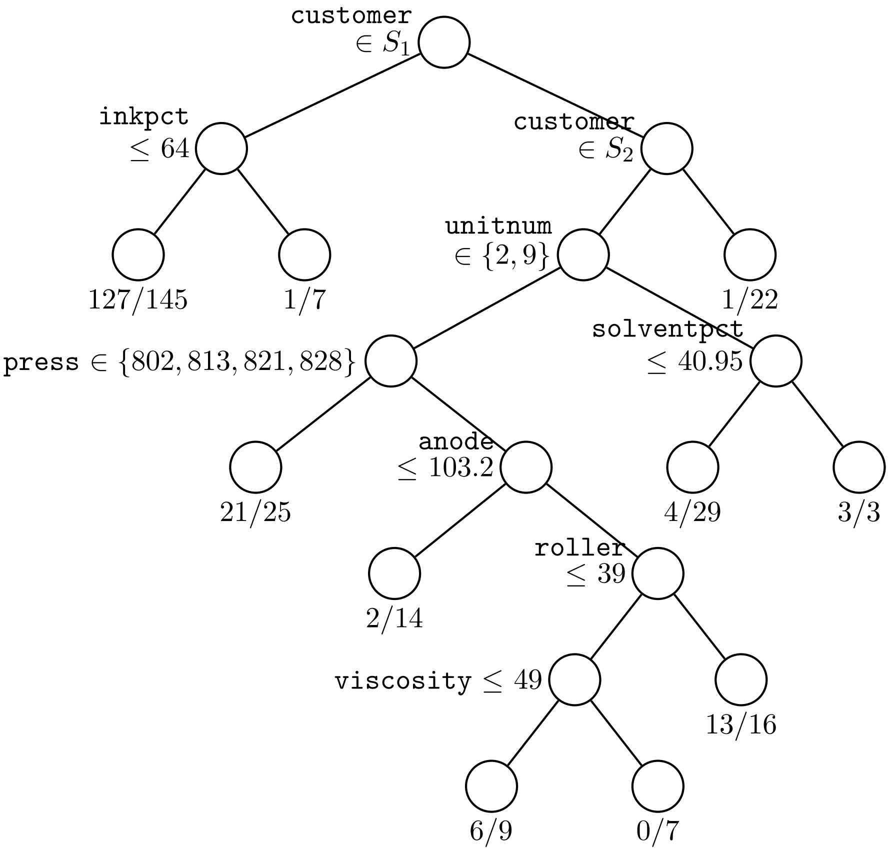

Figure 4: CART tree for the bands data without jobno. S_1 and S_2 are sets of customer names. The predicted value of $P(Y = 1)$ is given beneath each leaf node as the number of cases with $Y = 1$ divided by the node sample size. The cross-validation estimate of misclassification cost is 0.300.

we obtain the tree shown in Figure 4. It has 10 leaf nodes and splits first on customer which has $2^{50} - 1 \approx 10^{15}$ allowable splits. Finally, if we remove both jobno and customer, we get the tree in Figure 5. It splits first on press, which has $2^7 - 1 = 127$ allowable splits. The interesting question is whether the selection of jobno and customer in the first two trees is due to their importance in predicting Y or to bias from their having inordinately large numbers of splits. One answer can be found in the cross-validation estimates of misclassification costs of the trees. If jobno, customer, and press are indeed the three top predictors in order of importance, we would expect the estimated misclassification costs to increase as the variables are removed. But this does not happen. The estimates are 0.296, 0.300, and 0.271, respectively. This strongly suggests that selection bias is the culprit here.

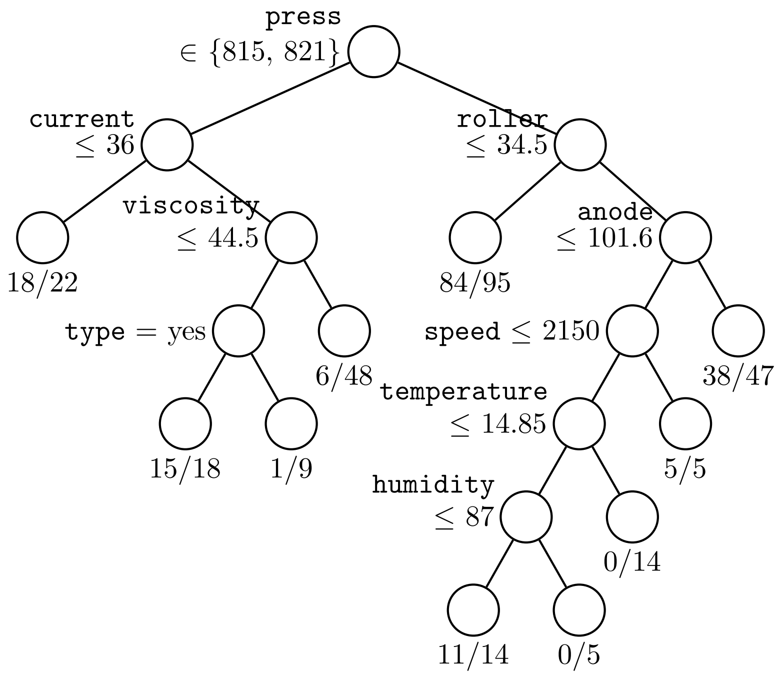

Figure 5: CART tree for the bands data without `jobno` and `customer`. The predicted value of $P(Y = 1)$ is given beneath each leaf node as the number of cases with $Y = 1$ divided by the node sample size. The cross-validation estimate of misclassification cost is 0.271.

Selection bias is always present in any algorithm that uses exhaustive search for variable selection. One way to ameliorate this is to include a weight factor in the search criterion to decrease the influence of variables that have many splits. This approach is problematic, however, for two reasons. First, there is no universal weighting formula—the weights must depend on the number and types of variables in the dataset as well as the sample size. Second, the computational complexity of recursively partitioning the data and using exhaustive search to find logistic regression leaf models is too overwhelming in real applications.

A practical approach is to break the task of split selection into two parts: first select the variable and then find the split values. This solution is used in the QUEST (Loh and Shih 1997) and CRUISE (Kim and Loh 2001) classification tree algorithms, where statistical tests are used for variable selection. It is applied to regression in the GUIDE algorithm (Loh 2002) by means of "curvature tests". Basically, a curvature test is a chi-square test of association for a two-way contingency table where the rows are determined by the signs of the residuals (positive versus non-positive) from a fitted regression model. If the variable is categorical, the columns are defined by the category names. Otherwise, if the variable is quantitative, its values are discretized into four interval groups using the sample quartiles as endpoints; the groups are then used as columns for the table. The idea is that if a model fits well, its residuals should have little or no association with the values of the predictor variable and the significance probability of the chi-square test should be approximately uniformly distributed. On the other hand, if the effect of the variable is not fully accounted for, the significance probability is expected to be small. Thus one way to choose a variable for splitting is to select the one with the smallest significance probability.

This technique is not immediately applicable to linear logistic regression. As mentioned previously, the value of Y determines the sign of the residual, $Y - \hat{p}$. Thus the chi-square test of association between the signs of the residuals and the values of X is the same as the test of association between Y and X. But the latter is independent of the model! Hence the test cannot distinguish between the linear and nonlinear effects of X. Simulation results to be presented later will show that if X_1 has a strong linear effect and X_2 has a weak quadratic effect, the test will erroneously select X_1 instead of X_2 with high probability.

To distinguish nonlinear from linear effects, we use a trend-adjusted chi-square test due to Cochran (1954) and Armitage (1955). Consider a $2 \times J$ contingency table with rows being the values of Y and columns being some grouped scores $X = x_j$, $j = 1, \ldots, J$. Let n_{ij} denote the number of cases with $Y = i$ $(i = 0, 1)$ and $X = x_j$. Then the sample size is $n = \sum_i \sum_j n_{ij}$. Denote the sample proportion of each cell by $p_{ij} = n_{ij}/n$. The conditional proportion of times that observations in column j have response i is $p_{i|j} = p_{ij}/p_{+j} = n_{ij}/n_{+j}$, where $n_{+j} = np_{+j} = \sum_i n_{ij}$. Let $\pi_{1|j}$ denote the conditional probability that $Y = 1$ given $X = x_j$. For the linear probability model

$$\pi_{1|j} = \alpha + \beta x_j \tag{2}$$

a least squares fit gives the fitted values $\hat{\pi}_{1|j} = p_{1+} + b(x_j - \bar{x})$, where $p_{1+} = n^{-1} \sum_j n_{1j}$, $\bar{x} = n^{-1} \sum_j n_{+j} x_j$ and

$$b = \sum\nolimits_j n_{+j}(p_{1|j} - p_{1+})(x_j - \bar{x}) \Big/ \sum\nolimits_j n_{+j}(x_j - \bar{x})^2.$$

Cochran (1954) and Armitage (1955) show that the Pearson chi-square statistic, χ^2, for testing independence can be decomposed as $\chi^2 = z^2 + \chi^2_L$, where

$$\begin{aligned} z^2 &= b^2 \sum\nolimits_j n_{+j}(x_j - \bar{x})^2/(p_{0+}p_{1+}) \\ \chi^2_L &= \sum\nolimits_j n_{+j}(p_{1|j} - \hat{\pi}_{1|j})^2/(p_{0+}p_{1+}). \end{aligned}$$

The statistic z^2 is called the Cochran-Armitage trend test statistic. If model (2) holds, z^2 has an asymptotic chi-square distribution with one degree of freedom. It tests for a linear trend in the proportions. The statistic χ^2_L is asymptotically chi-square distributed with $J - 2$ degrees of freedom. It tests for independence between X and Y after adjusting for any linear trend. We call χ^2_L the *trend-adjusted* chi-square statistic and use it to obtain a significance probability for each n-variable, after first discretizing its values into five groups at the sample quintiles.

Since categorical (c- and o-) variables are used for splitting the nodes only, no similar trend adjustment is necessary. Therefore we simply use the ordinary chi-square test between Y and the categorical variable. The null distribution of the statistic is approximated with a chi-square distribution with $C(t) - 1$ degrees of freedom, where $C(t)$ is the number of category values of the variable at node t. A similar method is used to treat s-variables, except that their values are first discretized into five groups at the sample quintiles.

We thus obtain a significance probability for each s-, n-, c- and o-variable. The variable with the smallest significance probability is selected to split the node. The following algorithm details the steps performed at each node.

Algorithm 2 Variable Selection Algorithm

1. *For each* n*-variable X, divide its values into five groups at the sample quintiles. Construct a 2×5 contingency table with the Y values as rows and the five groups as columns. Count the number of cases in each cell. If X is used in the linear logistic regression model fitted to the node, compute the trend-adjusted chi-square statistic; otherwise, compute the ordinary chi-square statistic. The column scores used in the trend-adjusted chi-square statistic are the sample means of the X values in their respective groups. The degrees of freedom for the with and without trend-adjusted statistics are* $(2-1)(5-1)-1 = 3$ *and* $(2-1)(5-1) = 4$*, respectively. Compute the corresponding significance probability.*

2. *Repeat the 2×5 contingency table construction procedure in step 1 for each* s*-variable X. Compute the significance probability of the ordinary chi-square statistic based on 4 degrees of freedom.*

3. *For each* c*- and* o*-variable X, construct a contingency table using the Y values as rows and the categories of X as columns. Compute the corresponding significance probability of the ordinary chi-square statistic based on $C - 1$ degrees of freedom, where C is the number of distinct values of X at the node.*

4. *Select the variable with the smallest significance probability to split the node.*

Figure 6 shows the result of applying the LOTUS(S) method to the bands data. The tree has only one split, on the variable press. This is the same as the variable selected as the top split in the CART tree in Figure 5. Note that current and roller, the linear predictors in the two leaf nodes in the LOTUS(S) tree are also chosen by the CART tree as the next most important variables. The most interesting difference between the two trees is, however, hidden—the LOTUS(S) tree is constructed using all the variables whereas the CART tree requires manual removal of jobno and customer.

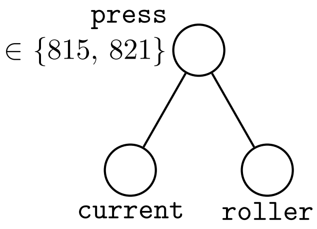

Figure 6: LOTUS(S) tree for the bands data. The best linear predictor is given beneath each leaf node.

4 SIMULATION EXPERIMENTS

There are two obvious sources of approximation error in Algorithm 2. First, the linear probability model (2) is at best a rough approximation to the linear logistic regression model $\text{logit}(\pi_{1|j}) = \alpha + \beta x_j$. Second, the chi-square distributions used in the computation of the significance probabilities are not exact. We report here the results of some simulation experiments to evaluate the effects of these approximations.

4.1 Selection bias and power

Our first experiment compares the selection bias and selection power of the trend-adjusted method versus the same method without the adjustment (called the "unadjusted method" henceforth). We employ five mutually independent predictor variables with marginal distributions and simulation models shown in Table 3. Predictors X_1 and X_5 have different degrees of discreteness, X_3 is symmetric while X_2 is skewed, and X_4 has a bimodal distribution. Predictor X_5 is used as an s-variable while the others are used as n-variables. A multiple linear logistic model is fitted to each node.

The Null, Linear, and Linlin models in Table 3 are used to evaluate the success of our bias correction method. In the Null model, Y is distributed independently of the five predictors. An unbiased variable selection procedure should therefore select each variable with equal probability of 1/5. The Linear, Linquad, Linlin, and Linlinquad models are designed to show the bias of the unadjusted method. The Jump, Quadratic, and Cubic models are nonlinear in X_1. They show how often each method detects the nonlinearity by choosing X_1 as split variable. The simulation results are based on 1000 iterations with sample size 500 in each iteration. This yields simulation standard errors of approximately 0.015.

Figure 7 shows bar graphs of the estimated selection probabilities. Selection bias is not apparent in both methods for the Null model. Despite differences in degrees of discreteness, skewness, and multi-modality of the predictor variable distributions, both methods select the predictors with roughly equal probabilities. For the Jump, Quadratic, and Cubic models, both methods select the correct variable X_1 most of the time. The selection power

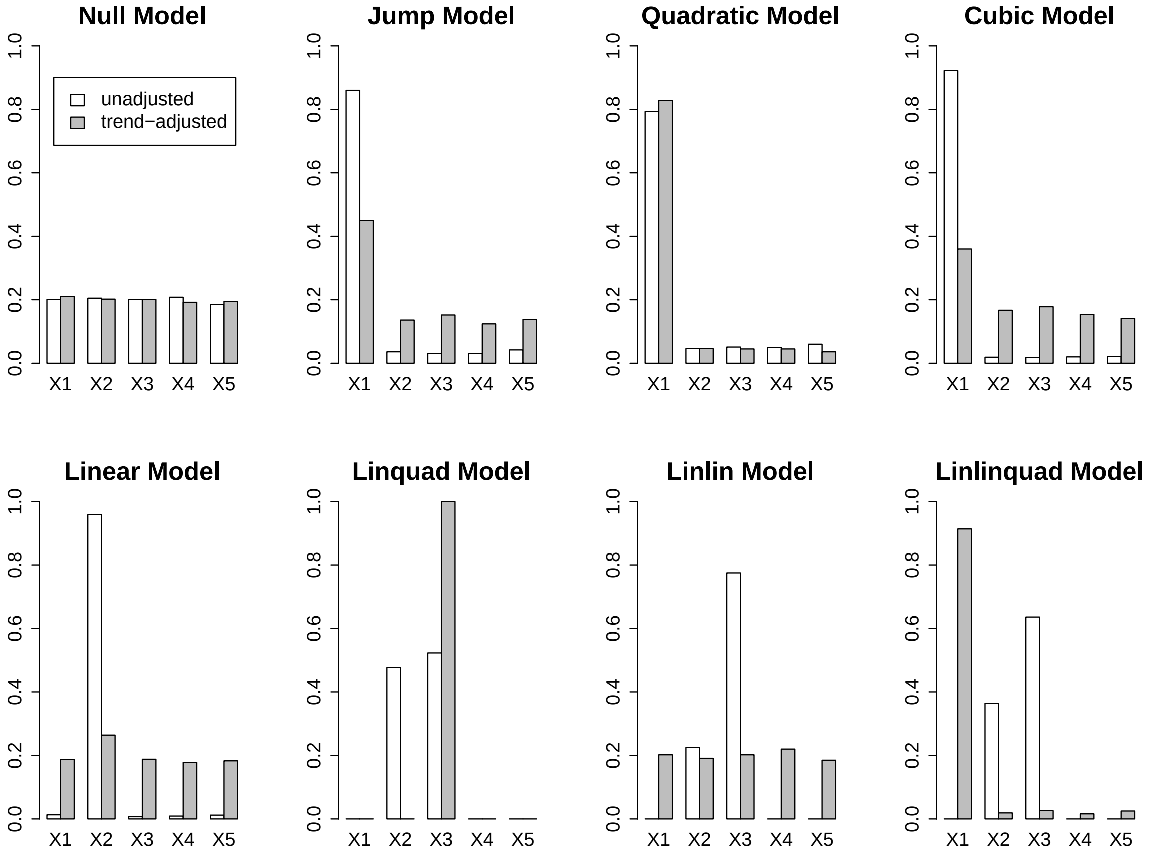

Figure 7: Simulated probabilities of variable selection for the unadjusted (white) and trend-adjusted (gray) methods under the models in Table 3. Simulation standard errors are approximately 0.015.

Table 3: Variables and models for selection power experiment. $N(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2; Exponential(μ) denotes an exponential distribution with mean μ; Uniform$\{a_1, \ldots, a_n\}$ denotes a uniform distribution on the integers $a_1, \ldots, a_n$.

Variable	Type	Distribution	Model	logit(p)
X_1	n	Uniform$\{-3, -1, 1, 3\}$	Null	0
X_2	n	Exponential(1)	Jump	$1 + 0.7I(X_1 > 0)$
X_3	n	$N(0, 1)$	Quadratic	$1 + 0.08X_1^2$
X_4	n	$0.5N(0, 1) + 0.5N(1, 1)$	Cubic	$1 + 0.02X_1^3$
X_5	s	Uniform$\{0, 1\}$	Linear	$1 + 0.8X_2$
			Linquad	$-1.5 + X_2 + X_3^2$
			Linlin	$-1 + X_2 + X_3$
			Linlinquad	$-1 - 0.1X_1^2 + X_2 + X_3$

of the trend-adjusted method may appear weaker than that of the unadjusted method at the Jump and Cubic models, but they perform equally well at the Quadratic model. One explanation is that the jump and cubic effects can partly be accounted for by a linear effect. Thus, when the trend-adjusted method is used, the strength of these nonlinear effects is reduced.

For the Linear and Linlin models, an unbiased selection procedure should select each variable with equal probability since the correct model is being fitted. Figure 7 shows that the unadjusted method does not possess this property. Because it does not take model fitting into consideration, it selects X_2 (in the Linear model) and X_2 or X_3 (in the Linlin model) much more frequently than the other variables. Note that this abnormality is not a concern (and is not apparent) if the tree after pruning has no splits. It becomes serious if the true model contains variables with linear and nonlinear effects. As the Linquad and Linlinquad models illustrate, the variables with linear effects may be selected more frequently than those with quadratic effects. Now pruning will yield a tree that either has splits on the wrong variables or has no splits at all. In contrast, the trend-adjusted method takes the linear effects into account and selects each variable with roughly equal probability in the Linear and Linlin models. In the Linquad and Linlinquad models, it ignores the linear effects and correctly selects the variable in the quadratic term most of the time.

4.2 Effect of variable type and collinearity

We performed two more simulation experiments to examine the effect of variable type and multicollinearity on the selection bias of LOTUS with different node models. The experiments employ three numerically ordered (X_1, X_2, X_3) and two nominal (X_4, X_5) variables. Three dependence structures are studied: (1) the "independence" case where the variables are mutually independent, (2) a "weak dependence" case where some of the variables are not independent, and (3) a "strong dependence" case where the correlation between X_2 and X_3 is 0.995. The distributions of the variables are given in Tables 4 and 5 and the joint distribution of X_4 and X_5 in Table 6. The response variable is independently and identically distributed Bernoulli with probability 0.5 in all cases.

Table 4: Marginal distributions of the X variables

Variable	Distribution
T	Discrete Uniform$\{-3, -1, 1, 3\}$
W	Exponential(1)
Z	Normal$(0, 1)$
U	Uniform$(0, 1)$
C_5	Discrete Uniform$\{1, 2, \ldots, 5\}$
C_{10}	Discrete Uniform$\{1, 2, \ldots, 10\}$

Table 5: Dependence structures of the X variables. The symbol $\lfloor \cdot \rfloor$ denotes the greatest integer function.

Variable	Type	Independence	Weak Dependence	Strong Dependence
X_1	Continuous	T	T	T
X_2	Continuous	W	W	W
X_3	Continuous	Z	$T + W + Z$	$W + 0.1Z$
X_4	Categorical	C_5	$\lfloor UC_{10}/2 \rfloor + 1$	$\lfloor UC_{10}/2 \rfloor + 1$
X_5	Categorical	C_{10}	C_{10}	C_{10}

The first experiment uses X_3 as an n-variable while the second employs it as an s-variable. The simulations are based on 1000 runs and sample size of 500 observations in each run, yielding simulation standard errors of approximately 0.015. The estimated selection probabilities are given in Table 7. All the entries lie within three simulation standard errors of 0.2 (the value for unbiased selection) regardless of mix of variable types, degree of multicollinearity and differences in node models. This shows that the unbiased selection

Table 6: Joint distribution of X_4 and X_5 in the weak and strong dependence situations

	X_5									
X_4	1	2	3	4	5	6	7	8	9	10
1	1/10	1/10	2/30	1/20	2/50	1/30	2/70	1/40	2/90	1/50
2			1/30	1/20	2/50	1/30	2/70	1/40	2/90	1/50
3					1/50	1/30	2/70	1/40	2/90	1/50
4							1/70	1/40	2/90	1/50
5									1/90	1/50

property of LOTUS is robust.

5 SPLIT POINT SELECTION

After X is selected to split a node, the next step is to find the split point (if X is ordered) or the split set (if X is nominal). For an ordered X, the most obvious way to choose a split of the form $X \leq c$ is to search for the c that minimizes the total deviance of the logistic regression models fitted to the two data subsets defined by the split. This is computationally prohibitive for logistic regression models. A much faster method uses the sample mean or median of X for c, but this may be ineffective if the true logit function is not smooth. As a compromise, LOTUS restricts the search to a set of sample quantiles of X. In the examples here, the search is over the sample 0.3, 0.4, 0.5, 0.6, and 0.7-quantiles of X at the node. The one that minimizes the total (logistic) deviance is selected.

If the selected X is a nominal variable, we need to find a set A of its values for a split of the form $X \in A$. Again, exhaustive search is computationally prohibitive. Instead, we limit the search to five candidates that are obtained by treating the problem as one of classification as follows. Let $\{a_1, a_2, \ldots, a_m\}$ denote the set of unique values taken by X at the node. Let $q(a)$ be the proportion of cases with $Y = 1$ among those in the node with $X = a$. Let b_1, b_2, $\ldots$, b_m be the ordered values of a according to $q(a)$, i.e., $q(b_1) \leq q(b_2) \leq \ldots \leq q(b_m)$. By a result in Breiman et al. (1984, p. 101), the value A^* that minimizes the sum of the Y-variance in the two data subsets created by the split $X \in A^*$ can be found by searching over the $m-1$ splits $X \in A_i$, where $A_i = \{b_1, \ldots, b_i\}$, $i = 1, \ldots, m-1$. Denote $A^* = \{b_1, \ldots, b_l\}$. For $j = \pm 1, \pm 2$ such that $1 \leq l + j \leq m - 1$, define $A^*_j = \{b_1, \ldots, b_{l+j}\}$. LOTUS evaluates the five nearby splits $X \in A$ with $A = A^*_{-2}$, A^*_{-1}, A^*, A^*_1, A^*_2 and selects the one that minimizes the sum of the logistic deviances in the two subnodes.

If none of the candidate splits for the selected variable is suitable (e.g., when one or

Table 7: Estimated probabilities of variable selection for LOTUS under different model and dependence situations

			Multiple linear logistic			Best simple linear logistic		
				Weak	Strong		Weak	Strong
Experiment	X_i	Type	Indep.	Dep.	Dep.	Indep.	Dep.	Dep.
	X_1	n	.189	.188	.213	.197	.183	.195
	X_2	n	.201	.219	.167	.206	.209	.182
1	X_3	n	.212	.188	.185	.193	.199	.186
	X_4	c	.208	.209	.220	.207	.199	.214
	X_5	c	.190	.196	.215	.197	.210	.223
	X_1	n	.196	.189	.208	.194	.173	.200
	X_2	n	.200	.214	.182	.204	.206	.175
2	X_3	s	.207	.196	.179	.219	.215	.192
	X_4	c	.211	.201	.218	.198	.199	.211
	X_5	c	.186	.200	.213	.185	.207	.222

more subnodes have zero variance or the parameter estimates fail to converge), LOTUS searches for the next best variable and its corresponding split values. This procedure is necessary to prevent premature termination of the tree growing process.

6 MISSING VALUES AND PRUNING

We have assumed thus far that the dataset has no missing values. If values are missing from the training sample, we use only the cases that are non-missing in the variables under consideration for regression modeling and split selection. That is, we fit a logistic regression model using only the cases that are non-missing in the designated set of regressors for the node model. For split selection, the significance probability of each candidate variable is computed from the non-missing cases of that variable. Once a split variable X is selected, the split point or split set is determined from the non-missing X-values. To send the cases with missing X-values through the split, we temporarily fill the missing values with the node sample X-mean (if X is an `n` or `s`-variable) or node sample X-mode (if X is a `c` or `o`-variable). After passing through the split, all imputed values are deleted and their missing status restored.

Missing values in a case to be predicted by a logistic regression tree present two problems: (1) how to pass the case through the splits for which the split variables are missing, and (2) how to compute the estimated logit value when one or more regressors in the node model are missing. For the first problem, we estimate any missing values with the training sample node mean (if X is numerical) or mode (if X is categorical) as in the previous paragraph. After the case passes through the splits, the imputed values are erased.

Our solution to the second problem depends on the model being fitted. If it is a multiple linear logistic model, we estimate the values of the missing regressors with the training sample node means. If it is a best simple linear logistic model and the required regressor is missing, we use the next best simple linear logistic model containing a non-missing regressor, where "best" refers to smallest deviance per degree of freedom.

LOTUS recursively partitions the sample space until there are too few observations in each partition to fit a non-trivial logistic regression model. It then employs the CART minimal cost-complexity technique to prune away branches, with "cost" being estimated predicted deviance. If an independent test sample is available, it may be used to estimate the latter; otherwise, ten-fold cross-validation is employed. The subtree with the lowest estimated predicted tree deviance is selected. The reader is referred to Breiman et al. (1984, Chap. 3) for details on the pruning procedure.

7 ACCURACY AND TRAINING TIME

We use thirty real datasets to compare LOTUS with stepwise linear logistic regression with respect to prediction accuracy and training time. Many of the datasets are from StatLib (`http://lib.stat.cmu.edu`), the University of California, Irvine, data repository (Blake and Merz 2000, UCI), and L. Torgo's website (`http://www.liacc.up.pt/~ltorgo/Regression/DataSets.html`, TRD). Some are derived from datasets whose response variables take more than two values. For example, the Letter-A and Letter-V datasets are derived from the UCI `letter` dataset by defining $Y = 1$ for the letter "A" and the class of vowels, respectively. Tables 8 and 9 contain brief descriptions of each dataset, including the source, number of observations, percentage of cases with missing values, number of predictor variables of each type, total number of variables for stepwise logistic regression, and percentage of cases with $Y = 1$.

We study both `LOTUS(S)` and `LOTUS(M)`. All quantitative variables are treated as `n`-variables. Stepwise linear logistic regression models (denoted by `LOGIST`) are fitted using the `glm` and `step` functions of S-PLUS (Venables and Ripley 1999). Each categorical variable in `LOGIST` is first converted into a set of $m - 1$ dummy variables, where m is the number of categories. Further, since there is no provision for ordinal variables in `glm`, they are treated as quantitative in `LOGIST`. The `step` function uses the *AIC* criterion (Akaike 1974), `AIC` $= -2$(maximized log-likelihood) $+$ 2(number of parameters). Starting with a full model, variables are dropped or added sequentially until `AIC` is minimized.

The predicted mean deviance (i.e., deviance divided by number of predicted cases) and the predicted mean square error are used as measures of accuracy. We estimate them by ten-fold cross-validation as follows. Each dataset is randomly divided into ten roughly equal-sized subsamples. One subsample (called *evaluation sample*) is removed in turn

Table 8: Brief descriptions of the datasets

ID	Name	Definition of event $Y = 1$	Source
1	Abalone	Age of abalone $\leq$ 10	UCI
2	Adult	Income > \$50,000	UCI
3	Ailerons	Control action on ailerons of F16 $\leq$ -0.0009	TRD
4	Annthyroid	Presence of hypothyroid disease	UCI
5	Birth	Low birth weight	Hosmer and Lemeshow (1989)
6	Boston	Median value of houses $\geq$ \$22,500	UCI
7	Breast	Presence of malignant tumor	UCI
8	Bupa	Presence of liver disorder	UCI
9	California	Median value of houses > \$130,000	StatLib
10	Car	American-made car	StatLib
11	Contracep	Contraceptive use in Indonesian women	UCI
12	Covertype	Lodgepole Pine versus Spruce-Fir	UCI
13	Cow	Survival of recumbent cows	Cook and Weisberg (1999, p. 467)
14	Credit	Approval of Australian credit card	UCI
15	German	Approval of German credit card	StatLog, UCI
16	Heart	Presence of heart disease	UCI
17	Housing8	Median price of house > \$33,200	TRD
18	Housing16	Median price of house > \$33,200	TRD
19	Letter-A	Letter ‘A’ versus non-‘A’	UCI
20	Letter-V	Vowel versus non-vowel	UCI
21	Otitis	Presence of otitis media in babies	Le (1998, pp. 233–238)
22	Pageblock	Text block versus non-text block	UCI
23	Pendigit	Digit ‘0’ versus others	UCI
24	Pima	Diabetes in Pima Indian women	UCI
25	Prostate	Tumor penetration of prostatic capsule	U. of Massachusetts dataset archive
26	Sick	Presence of hypothyroid disease	UCI
27	Teaching	Teaching assistant score in upper-third	Authors
28	Telecom	Telecommunication pole > 0	Weiss and Indurkhya (1995), TRD
29	Wage	Wage in upper-third of wage earners	Schafgans (1998)
30	Yeast	Cytosolic, nuclear, or mitochondrial site	UCI

Table 9: Characteristics of the datasets. The column labeled "#LOGIST parameters" refers to the number of variables after transformation of categorical variables to 0-1 variables; it is only relevant to LOGIST.

Dataset				Number of variables			#LOGIST	Percent
ID	Name	Size	%Missing	Quantitative	Nominal	Ordinal	parameters	$Y = 1$
1	Abalone	4177	0.0	7	1	0	9	65.4
2	Adult	48842	7.4	6	8	0	97	23.9
3	Ailerons	13750	0.0	12	0	0	12	42.4
4	Annthyroid	7200	0.0	6	15	0	21	92.6
5	Birth	189	0.0	2	4	2	9	31.2
6	Boston	506	0.0	12	1	0	13	41.9
7	Breast	699	2.3	9	0	0	9	34.5
8	Bupa	345	0.0	6	0	0	6	58.0
9	California	20640	0.0	8	0	0	8	71.4
10	Car	406	3.4	5	1	1	18	62.6
11	Contracep	1473	0.0	2	4	3	11	42.7
12	Covertype	495141	0.0	10	2	0	47	57.2
13	Cow	435	85.7	6	3	0	9	38.2
14	Credit	690	5.4	6	9	0	37	44.5
15	German	1000	0.0	7	13	0	48	30.0
16	Heart	303	2.0	6	7	0	18	45.9
17	Housing8	22784	0.0	8	0	0	8	49.8
18	Housing16	22784	0.0	16	0	0	16	49.8
19	Letter-A	20000	0.0	16	0	0	16	3.9
20	Letter-V	20000	0.0	16	0	0	16	19.4
21	Otitis	199	0.0	2	4	0	6	48.2
22	Pageblock	5473	0.0	10	0	0	10	89.8
23	Pendigit	10992	0.0	16	0	0	16	10.4
24	Pima	768	0.0	8	0	0	8	34.9
25	Prostate	380	1.1	4	3	0	9	40.3
26	Sick	3772	29.9	6	20	0	29	6.1
27	Teaching	324	0.0	2	4	0	73	33.3
28	Telecom	15000	0.0	48	0	0	48	37.7
29	Wage	3380	0.0	3	4	0	8	33.3
30	Yeast	1484	0.0	8	0	0	8	76.5

and the models are built by pooling the observations in the other nine subsamples (called *training sample*). The evaluation sample is then applied to each model to obtain the estimated predicted mean deviance

$$\mathrm{DEV} = -2n^{-1}\sum_{i=1}^{n}[y_i \log(\hat{p}_i/y_i) + (1-y_i)\log\{(1-\hat{p}_i)/(1-y_i)\}]$$

and estimated predicted mean square error $\mathrm{MSE} = n^{-1}\sum_{i=1}^{n}(y_i - \hat{p}_i)^2$. Here y_i and $\hat{p}_i$ are the response and predicted probability for the ith observation in the evaluation set and n is its sample size. The final estimates of predicted mean deviance and mean square error are obtained by averaging the ten DEV and MSE values, respectively.

For `LOGIST`, cases with missing predictor values in the training sample are omitted. Further, in the computation of its DEV and MSE values, any missing quantitative or categorical values in the evaluation samples are replaced with their training sample means or modes, respectively. Categorical values that appear in an evaluation sample but not in the training sample are treated as missing values. `LOGIST` is programmed in S-PLUS 2000 Professional Edition and LOTUS in Fortran 90. The computations for all except one dataset are obtained on a Windows XP Pentium III 700Mhz PC with 256MB of memory. Those for dataset 12 are from a Windows XP Pentium IV 2.4Ghz PC with 1GB of memory, because S-PLUS needed more memory.

Table 10 gives the results. The last row of the table shows that by both accuracy measures, `LOTUS(M)` is better on average than `LOTUS(S)` which is turn is better than `LOGIST`. To find out whether the differences are statistically significant, we follow Lim et al. (2000) and Kim and Loh (2001) and fit a mixed effects model separately to the mean deviance and the mean square error numbers in the table. The methods are treated as fixed effects and the datasets and the dataset-method interactions as random effects. The hypothesis of no method effects is strongly rejected in both cases—the significance probabilities are 0.003 for mean deviance and 0.01 for mean square error. Application of 90% simultaneous confidence intervals based on Tukey's studentized range (Miller 1989, pp. 37–48) reveals the difference in mean deviance between `LOTUS(S)` and `LOTUS(M)` to be not statistically significant, but both are significantly better than `LOGIST`. A similar analysis of the mean square error results shows that only `LOTUS(M)` is significantly better than `LOGIST`.

How much do `LOTUS(S)` and `LOTUS(M)` improve upon the prediction accuracy of `LOGIST` in percentage terms? Figure 8 shows barplots of the estimated predicted mean deviance of `LOTUS(S)` and `LOTUS(M)` relative to that of `LOGIST`. (The graph for predicted mean square error is not shown because it is very similar.) A method is more accurate than `LOGIST` at a given dataset if its bar has length less than 1. `LOTUS(S)` and `LOTUS(M)` are better than `LOGIST` in eighteen and twenty-three, respectively, out of thirty datasets. The average relative mean deviances for `LOTUS(S)` and `LOTUS(M)` are 0.916 and 0.863, respectively. Thus on average, `LOTUS(S)` and `LOTUS(M)` reduce the mean deviance of `LOGIST` by nine and sixteen percent, respectively. Ninety-five percent confidence intervals for the average

Table 10: Ten-fold cross-validation estimates of predicted mean deviance and mean square error

Dataset		Mean Deviance			Mean Square Error		
ID	Name	LOGIST	LOTUS(S)	LOTUS(M)	LOGIST	LOTUS(S)	LOTUS(M)
1	Abalone	0.915	0.953	0.864	0.149	0.158	0.141
2	Adult	0.643	0.658	0.638	0.103	0.105	0.102
3	Ailerons	0.549	0.635	0.531	0.086	0.099	0.083
4	Annthyroid	0.266	0.220	0.210	0.026	0.023	0.025
5	Birth	1.255	1.224	1.233	0.214	0.210	0.212
6	Boston	0.651	0.722	0.638	0.094	0.110	0.092
7	Breast	0.206	0.320	0.205	0.027	0.040	0.027
8	Bupa	1.250	1.306	1.254	0.215	0.224	0.212
9	California	0.649	0.478	0.485	0.101	0.071	0.071
10	Car	0.545	0.538	0.585	0.089	0.084	0.097
11	Contracep	1.218	1.156	1.134	0.210	0.195	0.191
12	Covertype	0.953	0.813	0.673	0.156	0.130	0.104
13	Cow	1.691	1.160	1.285	0.244	0.198	0.200
14	Credit	0.912	0.709	0.678	0.106	0.106	0.101
15	German	1.035	1.086	1.091	0.169	0.181	0.182
16	Heart	0.828	1.071	0.956	0.125	0.165	0.153
17	Housing8	0.945	0.786	0.731	0.147	0.124	0.115
18	Housing16	0.941	0.799	0.720	0.145	0.125	0.111
19	Letter-A	0.071	0.067	0.041	0.008	0.007	0.005
20	Letter-V	0.868	0.392	0.359	0.139	0.056	0.045
21	Otitis	1.265	1.298	1.303	0.218	0.224	0.230
22	Pageblock	0.304	0.206	0.281	0.039	0.024	0.037
23	Pendigit	0.115	0.055	0.057	0.012	0.006	0.008
24	Pima	0.974	1.021	0.965	0.158	0.166	0.156
25	Prostate	1.071	1.129	1.084	0.180	0.191	0.182
26	Sick	0.212	0.154	0.183	0.027	0.017	0.022
27	Teaching	1.199	1.238	1.236	0.205	0.214	0.214
28	Telecom	0.655	0.279	0.212	0.098	0.039	0.029
29	Wage	0.923	0.915	0.890	0.149	0.146	0.142
30	Yeast	0.562	0.497	0.508	0.081	0.069	0.070
	Average	0.789	0.729	0.701	0.124	0.117	0.112

relative mean deviance for LOTUS(S) and LOTUS(M) are $(0.827, 1.005)$ and $(0.787, 0.938)$, respectively. A two-sided paired t-test of the relative mean deviances of LOTUS(S) and LOTUS(M) has a significance probability of 0.0504.

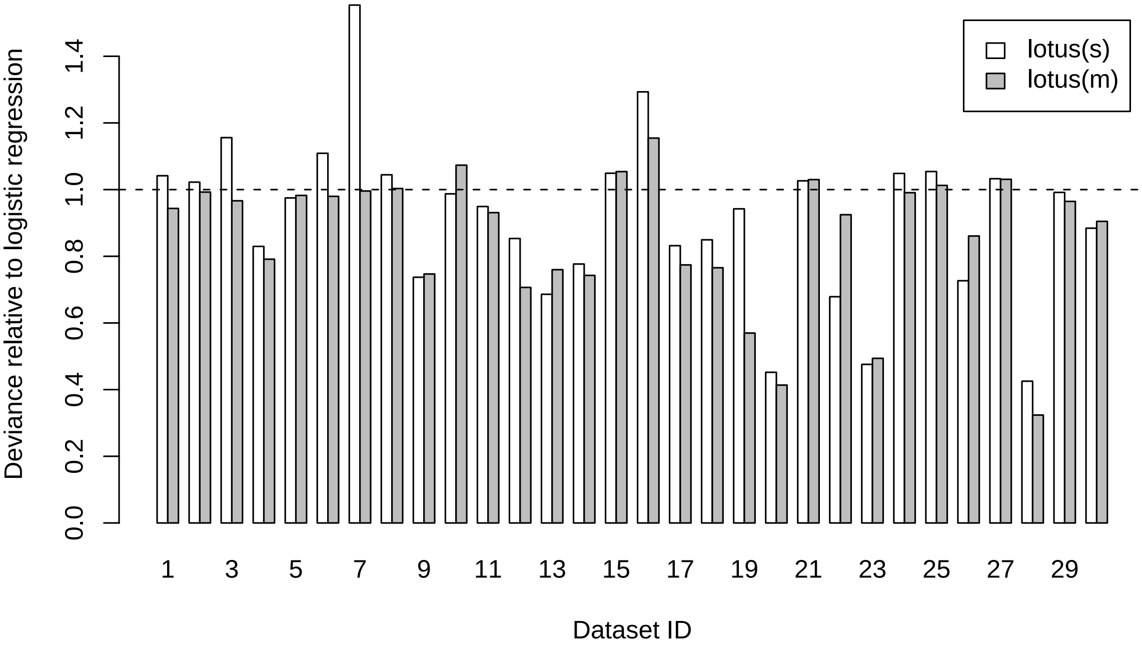

Figure 8: Predicted mean deviance of LOTUS(S) and LOTUS(M) relative to that of LOGIST

Table 11 gives the total execution times over the ten cross-validation runs. LOGIST is slowest on six and fastest on twenty-one datasets. LOTUS(S) is slowest on on fifteen and fastest on one. LOTUS(M) is slowest on nine and fastest on six. The speed of each method is affected by different factors. Recall that LOGIST first converts each categorical variable into a set of dummy variables and then uses stepwise variable selection. This puts it at a disadvantage when there are categorical variables with many levels, because LOTUS uses neither categorical variables nor stepwise variable selection in its node modeling. A good example is dataset 2, which has 6 quantitative and 8 categorical variables. After conversion to dummy variables, the effective number of variables for LOGIST is 97 (see Table 9). As a result, the execution time for LOGIST is twice as long as that for the two LOTUS methods.

Figure 9 plots the execution times of LOTUS(S) and LOTUS(M) relative to that of LOGIST against the sample size of each dataset. The relative times range from 0.13 to 28.7, increasing roughly linearly at the rate of the square root of sample size. The dependence on sample size is mainly due to large datasets requiring correspondingly large trees to be grown and then pruned. LOTUS(S) and LOTUS(M) are both faster than LOGIST for sample sizes less than 350. LOTUS(S) takes longer than LOTUS(M) for sample sizes up to about

Table 11: Total execution time of the cross validation runs for each dataset-method pair. The letters ‘s’, ‘m’ and ‘h’ denote seconds, minutes and hours, respectively. The times for dataset 12 are obtained on a faster machine than for the other datasets.

ID	Name	LOGIST	LOTUS(S)	LOTUS(M)	ID	Name	LOGIST	LOTUS(S)	LOTUS(M)
1	Abalone	41s	15.0m	8.4m	16	Heart	19s	14s	7s
2	Adult	4.4h	2.2h	2.4h	17	Housing8	6.8m	1.1h	1.2h
3	Ailerons	4.4m	54.2m	1.1h	18	Housing16	7.1m	2.1h	3.4h
4	Annthyroid	3.5m	12.4m	9.2m	19	Letter-A	9.5m	55.8m	1.5h
5	Birth	11s	5s	4s	20	Letter-V	8.1m	1.4h	2.2h
6	Boston	18s	1.4m	25s	21	Otitis	15s	2s	2s
7	Breast	16s	40s	19s	22	Pageblock	1.0m	16.7m	18.0m
8	Bupa	6s	34s	21s	23	Pendigit	3.2m	27.7m	59.2m
9	California	5.7m	1.2h	1.3h	24	Pima	13s	1.8m	1.3m
10	Car	13s	20s	12s	25	Prostate	13s	23s	13s
11	Contracep	35s	58s	46s	26	Sick	2.7m	4.8m	3.2m
12	Covertype	40.5m	17.0h	16.1h	27	Teaching	26s	10s	9s
13	Cow	15s	32s	11s	28	Telecom	9.7m	1.8h	3.8h
14	Credit	43s	40s	18s	29	Wage	1.2m	4.5m	2.9m
15	German	49s	1.8m	1.1m	30	Yeast	18s	3.6m	1.9m

5000. For larger datasets, LOTUS(M) almost always takes longer.

8 INSURANCE EXAMPLE

We now use a dataset from the Dutch insurance industry to illustrate how LOTUS(S) can identify and produce an intelligible profile of the prospective customers from a database. The dataset comes from the UCI Knowledge Discovery in Databases (KDD) archive. It was used to compare data mining packages at the Computational Intelligence and Learning (CoIL) Challenge 2000 competition (van der Putten, de Ruiter and van Someren 2000). The training set contains 5822 customer records. Each record consists of 85 predictor variables, representing socio-demographic data derived from area ZIP codes (variables 1–43) and product usage data (variables 44–85). All customers living in areas with the same ZIP code have the same socio-demographic values. Two variables are categorical: customer subtype (mostype, forty categories) and customer main type (moshoofd, ten categories). The rest are quantitative. The Y variable, caravan, equals 1 if a customer owns a caravan insurance policy, and 0 otherwise. Of the 5822 customers in the training set, only 348 (or six percent) own caravan insurance policies. A test set containing 4000 customer records is also given. The two goals of the CoIL challenge were: (1) to build a model from the

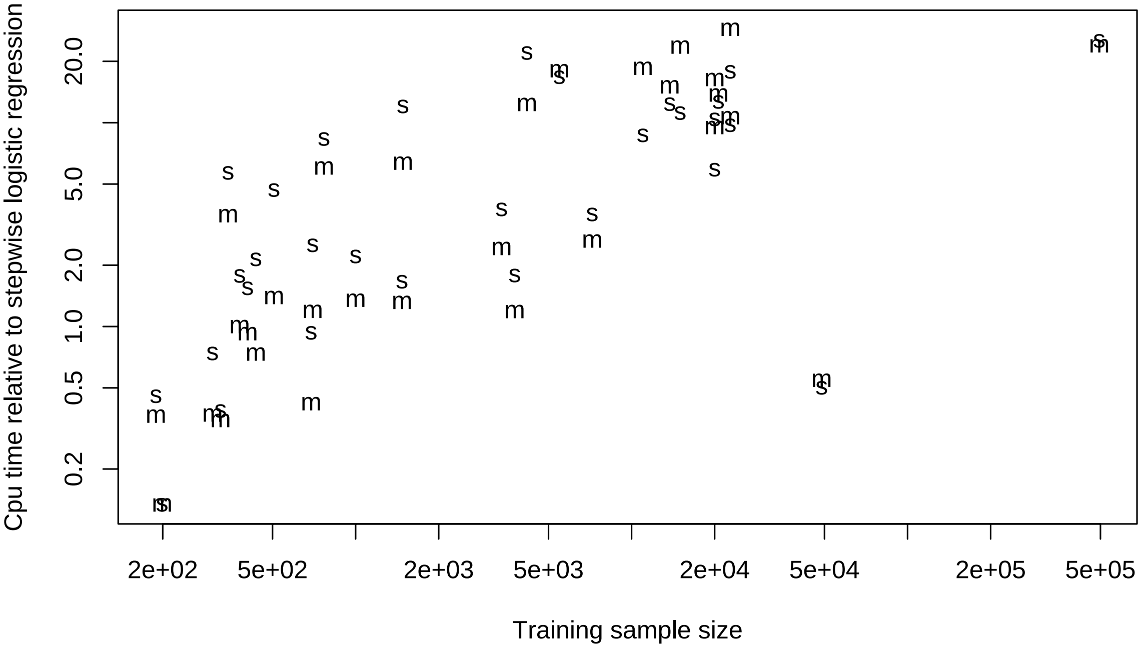

Figure 9: Plot of training time relative to `LOGIST` versus sample size. `LOTUS(S)` and `LOTUS(M)` are indicated by the plot symbols '`s`' and '`m`', respectively. The axes are in logarithmic scale.

5822 training records and use it to find the top 20% of customers in the test set who are most likely to own caravan insurance policies and (2) to provide insight into why some customers have caravan insurance policies and how they differ from other customers.

There are 238 caravan policy owners in the test set. If we randomly select twenty percent, we would expect to get about 48 of them. Of the 43 participants in the CoIL Challenge, the top two solutions use naive Bayes methods to correctly identify 121 and 115 policy owners. The third best solution uses an evolutionary algorithm to correctly identify 112 policy owners. Other solutions employed neural networks, genetic programming, fuzzy classifiers, classification and regression trees, support vector machines, and inductive logic programming. The worst solution correctly identifies only 37 caravan policy owners. A marketing expert who served as a contest judge remarked that "almost all entries lacked a good description in words: participants seemed to forget that most marketeers find it difficult to read statistics (and understand them!)" (van der Putten et al. 2000).

Since caravan policy owners make up only six percent of the training set, it is difficult for many classification algorithms to beat the simple rule that predicts every customer as a caravan policy non-owner. CART, for example, yields a trivial tree with no splits.

We can force the tree to split by not pruning, but this runs the risk of increasing the misclassification rate. Alternatively, we can employ unequal misclassification costs, but then we face the problem of what cost ratio to use and the possibility that different ratios lead to different tree structures. Cost ratios of 2:1, 10:1, and 20:1 yield trees with one, four, and two leaf nodes, respectively. The tree with cost ratio 20:1 splits on contribution to car policies (`ppersaut`), giving a $\hat{p}$ of 86/3459 if `ppersaut` ≤ 5, and 262/2363 otherwise. This model is too crude, however, for identifying the caravan policy owners because there are 1617 test cases with `ppersaut` > 5.

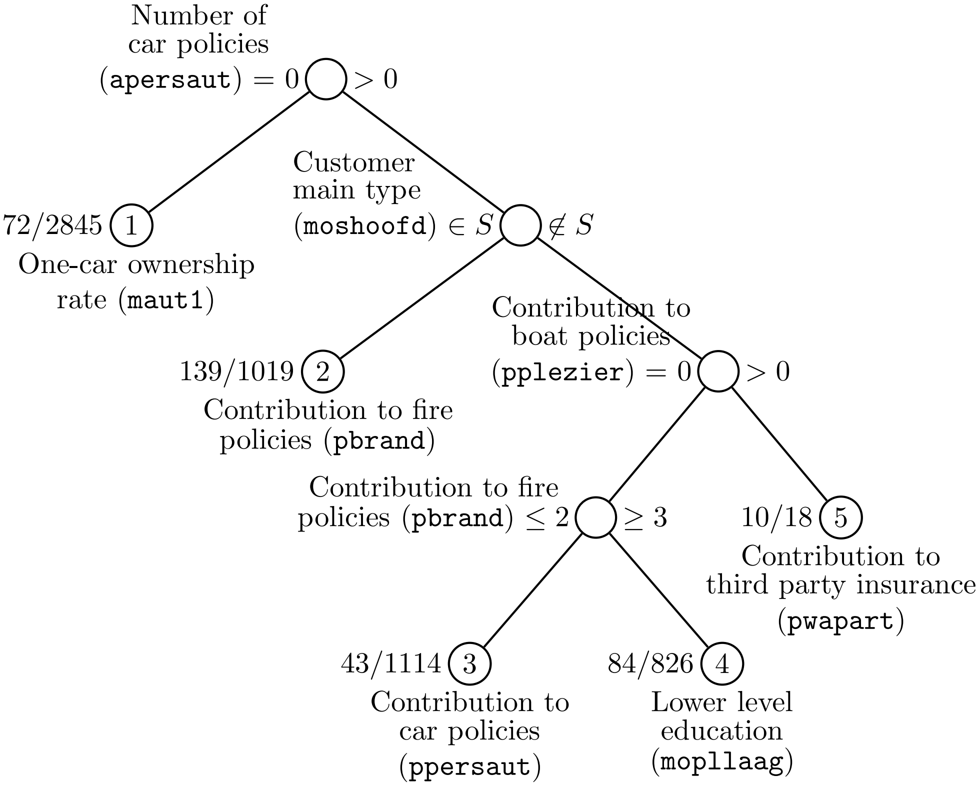

Figure 10: `LOTUS(S)` tree for the insurance data. At each leaf node are given the name of the selected regressor and the number of caravan policy holders divided by the training sample size. The set $S = \{$"successful hedonists," "driven growers," "average family"$\}$.

A more refined solution that is also easy to interpret is the five-leaf `LOTUS(S)` model displayed in Figure 10. The predictor variables chosen at the splits are number of car policies (`apersaut`), customer main type (`moshoofd`), contribution to boat policies (`pplezier`), and contribution to fire policies (`pbrand`). Three other variables also appear as regressors in the leaf nodes: one-car ownership rate (`maut1`), contribution to car policies (`ppersaut`), lower level education (`mopllaag`), and contribution to third party insurance (`pwapart`).

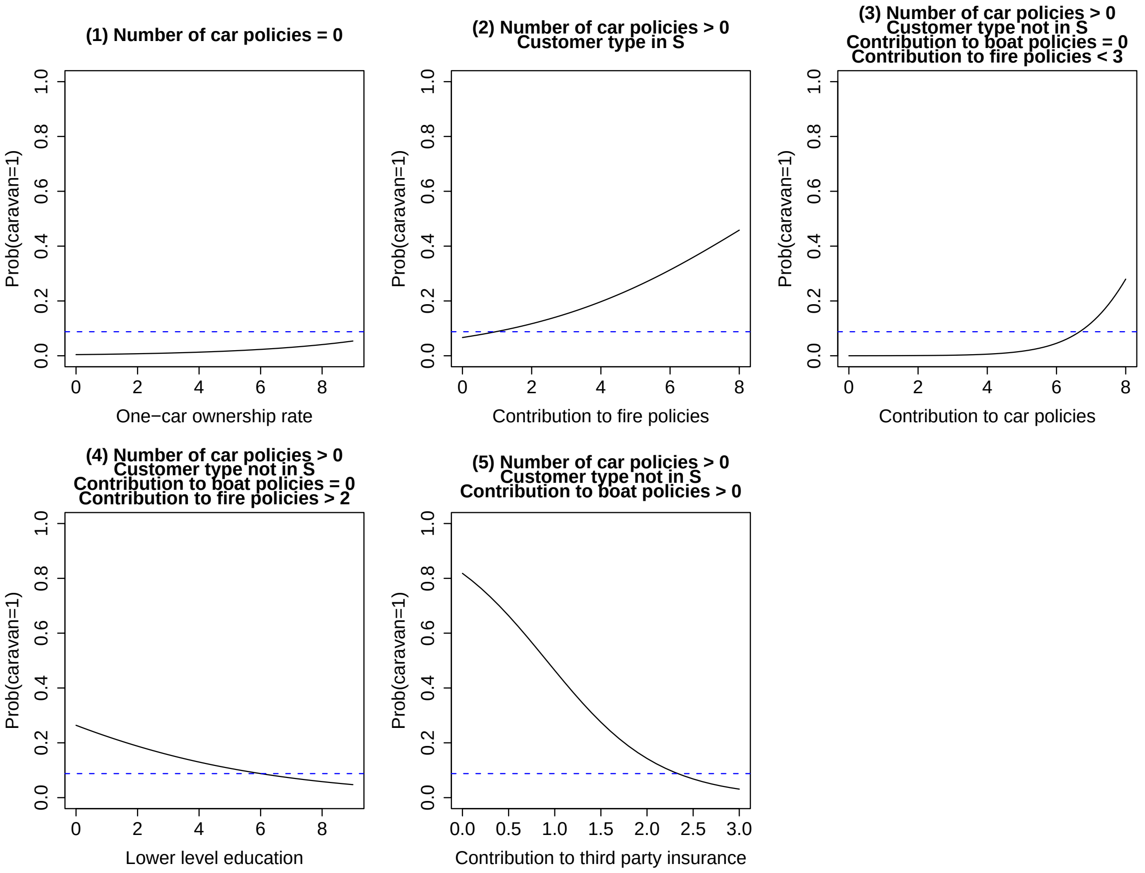

Figure 11: Fitted logistic regression functions in the leaf nodes of the `LOTUS(S)` tree in Figure 10. The dotted line marks the threshold value for identification of caravan policy owners in the test sample.

Plots of the $\hat{p}$ functions in the leaf nodes are shown in Figure 11. Theoretically, the top 800 (i.e., twenty percent) of the test cases most likely to own caravan policies are those for which $\hat{p} \geq \hat{p}_0$, where $\hat{p}_0 = 0.0877$ is the 0.8-quantile of the $\hat{p}$-values for the test set. The value of $\hat{p}_0$ is indicated by a dotted line in each plot. The Figure allows us to characterize the customers who own caravan policies as follows. Leaf node 1 customers do not own car policies and are least likely to own caravan policies. Leaf node 5 customers are most likely to own caravan policies, but they comprise only 0.3% of the training sample. They have one or more car policies, contribute to boat policies at level 1 or higher, and are not of type S, where S is the set of categories "successful hedonists," "driven growers," and "average family." The most likely places to find caravan policy owners are in leaf nodes 2 and 4, which contain approximately one-sixth and one-seventh, respectively, of the training sample. The customers in node 2 own one or more car policies and belong to type S. Those in node 4 also own one or more car policies, contribute to fire policies at level 3 or higher, but are not of type S and do not contribute to boat policies. Leaf node 3 contains about one-fifth of the training sample, but only thirteen customers have $\hat{p} > \hat{p}_0$ and none has $\hat{p} = \hat{p}_0$. These customers own one or more car policies, are not of type S, do not contribute to boat policies, contribute to fire policies at level 2 or lower, and contribute to car policies at level 7 or higher.

Now we can select the 800 test cases most likely to own caravan policies. First we include every test case for which $\hat{p} > \hat{p}_0$. This nets 0, 415, 9, 284, and 7 cases from leaf nodes 1–5, respectively. The numbers of caravan policy owners among them are 0, 70, 0, 31, and 1, respectively. This yields 715 test customers of which 102 own caravan policies. There are 123 test customers with $\hat{p} = \hat{p}_0$, all belonging to leaf node 4 and having lower level education 6. Among them, 12 own caravan policies. Depending on which 85 of the 123 test cases are selected, we can have between 102 and 114 caravan policy owners.

A simple way to break the ties is to apply stepwise logistic regression to the tied observations. That is, we fit a stepwise logistic regression model (using the S-PLUS `step` function and starting with a constant model) to the 166 training cases in leaf node 4 that have $\hat{p} = \hat{p}_0$. The estimated probabilities from this model induce an ordering of the 123 tied test cases. The top 81 cases according to this ordering contain 10 caravan policy owners. The next 5 cases are tied and have 1 caravan policy owner among them. Thus we select the top 81 test cases and randomly choose 4 from the 5 ties. This yields a total of 112 or 113 caravan policy owners, with probabilities 0.2 and 0.8, respectively.

For comparison, stepwise logistic regression applied to the entire training sample of 5822 cases correctly identifies 110 test cases as caravan policy owners. Thus it has about the same accuracy as our method. On the other hand, the stepwise logistic model is much harder to interpret because it employs twenty predictor variables. Finally, we note that although our predicted number of 112 or 113 is lower than the 121 found by the naive Bayes method, the difference is not statistically significant because the standard errors of the predictions are approximately equal to 10.

9 CONCLUSION

Linear logistic regression is a well-understood and powerful technique for obtaining probability estimates in binary regression problems. Its main weaknesses are difficulties in model fitting and interpreting the regression coefficients when there are many predictor variables. Classification trees solve this problem by moving the model complexity entirely to the tree structure. But because they yield piecewise-constant estimates of p, large tree structures are often required to match the prediction accuracy and fine granularity of logistic regression models. Very large trees, however, are also very difficult to comprehend.

Logistic regression trees try to combine the prediction accuracy of logistic models with the interpretability of tree structures. By allowing the model complexity to be shared between a tree structure and a set of logistic regression node models, the user can balance tree structure complexity with node model complexity. For example, if multiple linear logistic regression is employed to model the data in the nodes, the tree may be kept short. If a richer tree structure is desired, simple linear logistic models may be fitted to the nodes. A clear advantage of a simple linear logistic model is that the regression coefficient can be interpreted directly without worry of collinearity and other complications.

Since selection bias can cause a tree structure to suggest incorrect conclusions about the effects of variables, special attention is paid to overcoming this problem in the LOTUS algorithm. Like the GUIDE linear regression tree method, LOTUS does this by separating the task of variable selection from that of split point selection. Unlike GUIDE, however, LOTUS cannot use the patterns of signs of the residuals from a fitted model for variable selection. This is because the signs of the residuals are independent of the fitted model. Instead LOTUS uses a trend-adjusted chi-square test to allow for linear effects in the model. A welcome side-effect of separating variable selection from split point selection is a substantial reduction in computation.

Besides model interpretation, prediction accuracy and computation speed are often equally important in practical applications. The results from the empirical study of real datasets in Section 7 show that LOTUS is on average more accurate than stepwise logistic regression. It is faster than the latter for sample sizes up to around 350 and is not more than thirty times slower for sample sizes up to a half million. Thus it is accurate and sufficiently fast for most practical applications.

Compiled binaries (for Windows, Linux, Sun Solaris and Digital Unix) and a user manual of the LOTUS computer program may be obtained from `http://www.stat.nus.edu.sg/~kinyee/lotus.html`.

ACKNOWLEDGEMENTS

The authors are indebted to two anonymous referees whose comments and suggestions contributed to many improvements in the manuscript. Chan's work was supported by National University of Singapore grant R-155-000-024-112. Loh's work was partially supported by U.S. Army Research Office grant DAAD19-01-1-0586.

REFERENCES

Akaike, H. (1974), "A new look at statistical model identification," *IEEE Transactions on Automatic Control*, AU-19, 716–722.

Allison, P. D. (1999), *Logistic Regression Using the SAS System: Theory and Application*, Cary, NC: SAS Institute, Inc.

Armitage, P. (1955), "Tests for linear trends in proportions and frequencies," *Biometrics*, 11, 375–386.

Blake, C. and Merz, C. J. (2000), "UCI Repository of Machine Learning Databases," Technical Report, Department of Information and Computer Science, University of California, Irvine (`http://www.ics.uci.edu/~mlearn/MLRepository.html`).

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984), *Classification and Regression Trees*, Belmont, California: Wadsworth.

Chaudhuri, P., Huang, M.-C., Loh, W.-Y. and Yao, R. (1994), "Piecewise-polynomial regression trees," *Statistica Sinica*, 4, 143–167.

Chaudhuri, P., Lo, W.-D., Loh, W.-Y. and Yang, C.-C. (1995), "Generalized regression trees," *Statistica Sinica*, 5, 641–666.

Clark, L. A. and Pregibon, D. (1992), "Tree-based models," in *Statistical Models in S*, J. M. Chambers and T. J. Hastie (eds.), 377–419, Pacific Grove: Wadsworth.

Cochran, W. G. (1954), "Some methods of strengthening the common χ^2 tests," *Biometrics*, 10, 417–451.

Cook, R. D. and Weisberg, S. (1999), *Applied Regression Including Computing and Graphics*, New York: Wiley.

Hosmer, D. W. and Lemeshow, S. (1989), *Applied Logistic Regression*, New York: Wiley.

Kim, H. and Loh, W.-Y. (2001), "Classification trees with unbiased multiway splits," *Journal of the American Statistical Association*, 96, 598–604.

Le, C. T. (1998), *Applied Categorical Data Analysis*, New York: Wiley.

Lim, T.-S., Loh, W.-Y. and Shih, Y.-S. (2000), "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms," *Machine Learning*, 40, 203–228.

Loh, W.-Y. (2002), "Regression trees with unbiased variable selection and interaction detection," *Statistica Sinica*, 12, 361–386.

Loh, W.-Y. and Shih, Y.-S. (1997), "Split selection methods for classification trees," *Statistica Sinica*, 7, 815–840.

McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models*, 2nd ed., London: Chapman & Hall.

Miller, R. G., Jr. (1981), *Simultaneous Statistical Inference*, 2nd ed., New York: Springer.

Perlich, C., Provost, F., and Simonoff, J. (2003), "Tree induction vs. logistic regression: A learning-curve analysis," *Journal of Machine Learning Research*, 4, 211–255.

van der Putten, P., de Ruiter, M. and van Someren, M. (2000), "CoIL Challenge 2000 Tasks and Results: Predicting and Explaining Caravan Policy Ownership," *CoIL Challenge 2000: The Insurance Company Case*, P. van der Putten and M. van Someren (eds.), Amsterdam: Sentient Machine Research. Also a Leiden Institute of Advanced Computer Science Technical Report 2000-09, June 22, 2000.

Quinlan, J. R. (1992), "Learning with continuous classes," in *Proceedings of AI'92 Australian National Conference on Artificial Intelligence*, 343–348, Singapore: World Scientific.

Schafgans, M. M. A. (1998), "Ethnic wage differences in Malaysia: Parametric and semiparametric estimation of the Chinese-Malay wage gap," *Journal of Applied Econometrics*, 13, 481–504.

Steinberg, D. and Cardell, N. S. (1998), "The hybrid CART-Logit model in classification and data mining," *Eighth Annual Advanced Research Techniques Forum*, American Marketing Association, Keystone, Colorado.

Venables, W. N. and Ripley, B. D. (1999), *Modern Applied Statistics with S-PLUS*, 3rd ed., New York: Springer.

Weiss, S. M. and Indurkhya, N. (1995), "Rule-based machine learning methods for functional prediction," *Journal of Artificial Intelligence Research*, 3, 383–403.