

# A regression tree approach to identifying subgroups with differential treatment effects

Wei-Yin Loh

`loh@stat.wisc.edu`

Department of Statistics, University of Wisconsin  
Madison, WI 53706, U.S.A.

Xu He

`hexu@amss.ac.cn`

Academy of Mathematics and Systems Science  
Chinese Academy of Sciences, Beijing, China

Michael Man

`man_michael@lilly.com`

Eli Lilly and Company  
Indianapolis, IN 46285, U.S.A.

## Abstract

In the fight against hard-to-treat diseases such as cancer, it is often difficult to discover new treatments that benefit all subjects. For regulatory agency approval, it is more practical to identify subgroups of subjects for whom the treatment has an enhanced effect. Regression trees are natural for this task because they partition the data space. We briefly review existing regression tree algorithms. Then we introduce three new ones that are practically free of selection bias and are applicable to two or more treatments, censored response variables, and missing values in the predictor variables. The algorithms extend the GUIDE approach by using three key ideas: (i) treatment as a linear predictor, (ii) chi-squared tests to detect residual patterns and lack of fit, and (iii) proportional hazards modeling via Poisson regression. Importance scores with thresholds for identifying influential variables are obtained as by-products. And a bootstrap technique is used to construct confidence intervals for the treatment effects in each node. Real and simulated data are used to compare the methods.

**Key words:** Missing values; proportional hazards; selection bias.

## 1 Introduction

For many diseases, such as cancer, it is often difficult to find a treatment that benefits all patients. Current thinking in drug development is to find a subject subgroup, defined by

individual characteristics, that shows a large treatment effect. Conversely, if a treatment is costly or has potential negative side effects, there is interest to look for subgroups for which it treatment is ineffective. This problem of searching for subgroups with differential treatment effects is known as *subgroup identification* [9, 13, 20].

To fix ideas, suppose for the moment that the response variable  $Y$  is uncensored and the treatment variable  $Z$  takes values  $l = 1, 2, \dots, L$ . Let  $\mathbf{X}$  denote a vector of covariates. Given a subgroup  $S$  defined in terms of  $\mathbf{X}$ , let  $R(S) = \max_{i,j} |E(Y|Z = i, S) - E(Y|Z = j, S)|$  denote the effect size of  $S$ . The goal is to find the maximal subgroup with the largest value of  $R(S)$ , where the size of  $S$  is measured in terms of its probability of occurrence  $P(S)$ . If  $Y$  is subject to censoring, we replace the mean of  $Y$  by the log-hazard rate so that  $R(S)$  is the largest absolute log-hazard ratio between any two treatments.

Consider, for example, data from a randomized trial of the German Breast Cancer Study Group [27, 30] with 686 subjects where the response is recurrence-free survival time in days. The trial was designed as a  $2 \times 2$  factorial comparing 3 vs. 6 cycles of chemotherapy and presence vs. absence of hormone therapy (Tamoxifen). The breast cancer data contain, however, no information on the number of cycles of chemotherapy, presumably because it was previously found not significant [31]. Median follow-up time was nearly 5 years and 387 subjects did not experience a recurrence of the disease during the trial (54 percent censoring). The variables are hormone therapy (`horTh`: yes, no), age (21–80 years), tumor size (`tsize`: 3–120 mm), number of positive lymph nodes (`pnodes`: 1–51), progesterone receptor status (`progrec`: 0–2380 fmol), estrogen receptor status (`estrec`: 0–1144 fmol), menopausal status (`menostat`: pre, post), and tumor grade (`tgrade`: 1, 2, 3). A standard proportional hazards regression model shows that hormone therapy has a significant positive effect on survival, with and without adjusting for the covariates [12, 29]. Since hormone therapy has side effects and incurs extra cost, it is useful to find a subgroup where the treatment has little effect.

Parametric and semi-parametric models such as the proportional hazards model do not easily lend themselves to this problem. Besides, if there are more variables than observations, such as in genetic data, these models cannot be used without prior variable selection. Regression tree models are better alternatives, because they are nonparametric, naturally define subgroups, scale with the complexity of the data, and are not limited by the number of predictor variables.

Following medical practice [15], we call a variable *prognostic* if it provides information about the response distribution of an untreated subject. That is, it has marginal effects on the response but does not interact with treatment. Examples are age, family history of disease, and prior therapy. A variable is *predictive* if it defines subgroups of subjects who are more likely to respond to a given treatment. That is, it has interaction effects with the treatment variable. Figure 1 shows two regression tree models and the Kaplan-Meier curves in the terminal nodes of the trees. In the  $G_s$  model on the left, variable `pnodes` is prognostic: recurrence probability is reduced if `pnodes` > 3, with and without treatment. In the  $G_i$  model on the right, variable `progrec` is predictive: hormone therapy has little effect if `progrec` ≤ 21 and an enhanced effect otherwise.

The main goal of this article is to introduce the algorithms that yield the models in Figure 1 and to compare them against existing solutions, which are briefly reviewed in Sec. 2. Sec. 3 presents the new algorithms for uncensored response variables. Sec. 4 compares the selection bias and accuracy of the new and old methods and Sec. 5 proposes a bootstrap

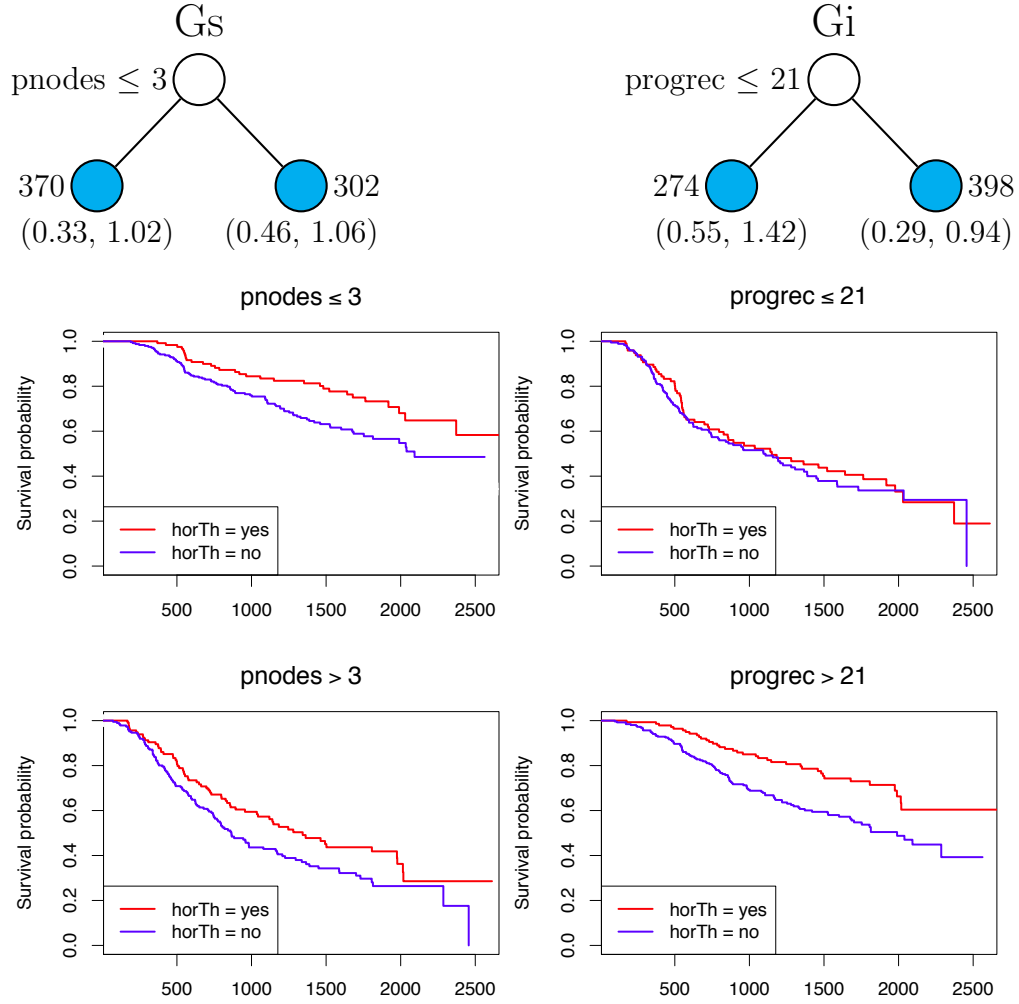


Figure 1:  $G_S$  (left) and  $G_i$  (right) tree models and Kaplan-Meier curves for breast cancer data. At each intermediate node, an observation goes to the left child node if and only if the displayed condition is satisfied. Sample sizes are beside terminal nodes; 95% bootstrap confidence intervals of relative risks for therapy versus no therapy are below nodes.

technique for computing confidence intervals of treatment and other effects in the subgroups. Sec. 6 extends the algorithms to censored survival data and Sec. 7 obtains importance scores for ranking the variables and thresholds for identifying the unimportant ones. Sec. 8 gives an application to a retrospective candidate gene study where there are large numbers of missing values and Sec. 9 concludes the article with some closing remarks.

## 2 Previous work

Let  $y_i$  and  $\mathbf{x}_i$  denote the survival time and covariate vector of subject  $i$ . Let  $s_i$  be an independent observation from some censoring distribution and let  $\delta_i = I(y_i < s_i)$  be the event indicator. The observed data vector of subject  $i$  is  $(y_i, \delta_i, \mathbf{x}_i)$ , where  $y_i = \min(y_i, s_i)$ ,  $i = 1, 2, \dots, n$ . Let  $\lambda(y, \mathbf{x})$  denote the hazard function at  $\mathbf{x} = (x_1, x_2, \dots, x_M)$ . The proportional hazards model specifies that  $\lambda(y, \mathbf{x}) = \lambda_0(y) \exp(\eta)$ , where  $\lambda_0(y)$  is a baseline hazard and  $\eta = \beta' \mathbf{x}$  is a linear function of the covariates.

Assuming that the treatment has two levels (denoted by  $z = 0, 1$ ), one approach [26] splits each node  $t$  into left and right child nodes  $t_L$  and  $t_R$  to maximize the Cox partial likelihood ratio statistic for testing  $H_0 : \lambda(y, \mathbf{x}) = \lambda_{0,t}(y) \exp\{\beta_0 z I(\mathbf{x} \in t)\}$  against  $H_1 : \lambda(y, \mathbf{x}) = \lambda_{0,t}(y) \exp\{\beta_1 z I(\mathbf{x} \in t_L) + \beta_2 z I(\mathbf{x} \in t_R)\}$ . A related approach, called *interaction trees* (IT) [33], chooses the split that minimizes the p-value from testing  $H_0 : \beta_3 = 0$  in the model  $\lambda(y, \mathbf{x}) = \lambda_{0,t}(y) \exp\{\beta_1 z + \beta_2 I(\mathbf{x} \in t_L) + \beta_3 z I(\mathbf{x} \in t_L)\}$ . If there is no censoring, the model is  $E(y) = \beta_0 + \beta_1 z + \beta_2 I(\mathbf{x} \in t_L) + \beta_3 z I(\mathbf{x} \in t_L)$ . Both methods employ the greedy search paradigm of evaluating all splits  $t_L = \{x_j \in S\}$  and  $t_R = \{x_j \notin S\}$  on every  $x_j$  and every  $S$ , where  $S$  is a half line if  $x_j$  is ordinal and is a subset of values if  $x_j$  is categorical. As a result, they are **computationally expensive and biased toward selecting variables that allow more splits**. Further, because  $\lambda_{0,t}(y)$  is a function of  $t$  and hence of  $\mathbf{x}$ , the tree models do not have proportional hazards and regression coefficients in different nodes cannot be compared.

The *virtual twins* (VT) method [13] is restricted to binary variables  $Y = 0, 1$ . It first generates a random forest [5] model to estimate the treatment effect  $\tau = P(Y = 1 | Z = 1) - P(Y = 1 | Z = 0)$  of each subject, using as split variables  $Z, X_1, \dots, X_M, ZX_1, \dots, ZX_M, (1 - Z)X_1, \dots, (1 - Z)X_M$ , with categorical variables converted first to dummy 0-1 variables. Then it uses RPART [34] to construct a classification or regression tree model to predict  $\tau$  for each subject and to obtain the subgroups. If a classification tree is used, the two classes are defined by the estimated  $\tau$  being greater or less than a pre-specified constant; if a regression tree is used, the subgroups are the terminal nodes with estimated  $\tau$  greater than a pre-specified constant. Although the basic concept is independent of random forest and RPART, their use gives VT all their weaknesses, such as variable selection bias and (with random forest) lack of a preferred way to deal with missing values. Further, VT is limited to binary  $Y$  and  $Z$ .

The *subgroup identification based on differential effect search* (SIDES) method [20] finds multiple alternative subgroups by identifying the best five (default) splits of each node that yield the most improvement in a desired criterion, such as the p-values of the differential treatment effects between the two child nodes, the treatment effect size in at least one child node, or the difference in efficacy and safety between the two child nodes. For each split, the procedure is repeated on the child node with the larger improvement. Heuristic and

resampling-based adjustments are applied to the p-values to control for multiplicity of splits and correlations among the p-values. The method appears to be most useful for generating candidate subgroups with large differential effects, but because only variables that have not been previously chosen are considered for splitting each node, the method may not be effective if the real subgroups are defined in terms of interval sets of the form  $\{a_j < X_j \leq b_j\}$ . The current implementation is limited to treatments with two levels.

Most methods can control the minimum node sample size so that the subgroups have sufficient numbers of observations. The *qualitative interaction tree* (QUINT) method [11] deals with this directly by optimizing a weighted sum of a measure of effect size and a measure of subgroup size. It looks for “qualitative interactions,” where one treatment performs better than another in one subgroup and worse in another subgroup. Similar to the above methods, QUINT finds the subgroups by searching over all possible splits on all predictor variables. Its current implementation is limited to ordinal  $X_i$ , uncensored  $Y$ , and binary  $Z$ .

### 3 Uncensored data

By evaluating all possible splits on all variables to optimize an objective function, each method (except possibly for SIDES) is biased toward selecting variables that allow more splits. This is due to an ordinal variable with  $k$  unique values yielding  $k - 1$  splits and a categorical variable with the same number of unique values yielding  $2^{k-1} - 1$  splits. As a result, **a variable that allows many splits has a greater chance to be selected than one with few splits**. Besides increasing the chance of spurious splits, the bias can undermine the credibility of the results. SIDES tries to control the bias with Bonferroni-type adjustments, but this can lead to over correction, as in the CHAID [16] classification tree algorithm, which is biased toward selecting variables with few splits.

The GUIDE algorithm [21, 23] overcomes this problem by using a two-step approach to split selection: first find the split variable and then search for the best split on the selected variable. The first step yields substantial computational savings, because there is no need to find the best splits on all the other variables. It also eliminates selection bias, at least in principle, by using **chi-squared tests to select the split variable**. QUEST [25], CRUISE [17], and CTREE [14] are other algorithms that employ significance tests for variable selection. In this section we extend GUIDE to subgroup identification for the case where  $Y$  is not censored.

#### 3.1 Gc: classification tree approach

This method requires that  $Y$  and  $Z$  are binary, taking values, 0, and 1, say. Then a classification tree may be used to find subgroups by defining the **class variable** as  $V = Y + Z \bmod 2$ :

$$V = \begin{cases} 0, & \text{if } \{Y = 1 \text{ and } Z = 1\} \text{ or } \{Y = 0 \text{ and } Z = 0\}, \\ 1, & \text{if } \{Y = 0 \text{ and } Z = 1\} \text{ or } \{Y = 1 \text{ and } Z = 0\}. \end{cases}$$

This is motivated by the observation that the subjects for which  $V = 0$  respond differentially to treatment and those for which  $V = 1$  do not. **Thus a classification tree constructed with  $V$  as the response variable will likely identify subgroups with differential treatment effects.**

Although any classification tree algorithm may be used, we use GUIDE [23] here because it does not have selection bias, and call it the Gc method (“c” for classification).

### 3.2 Gs and Gi: regression tree approach

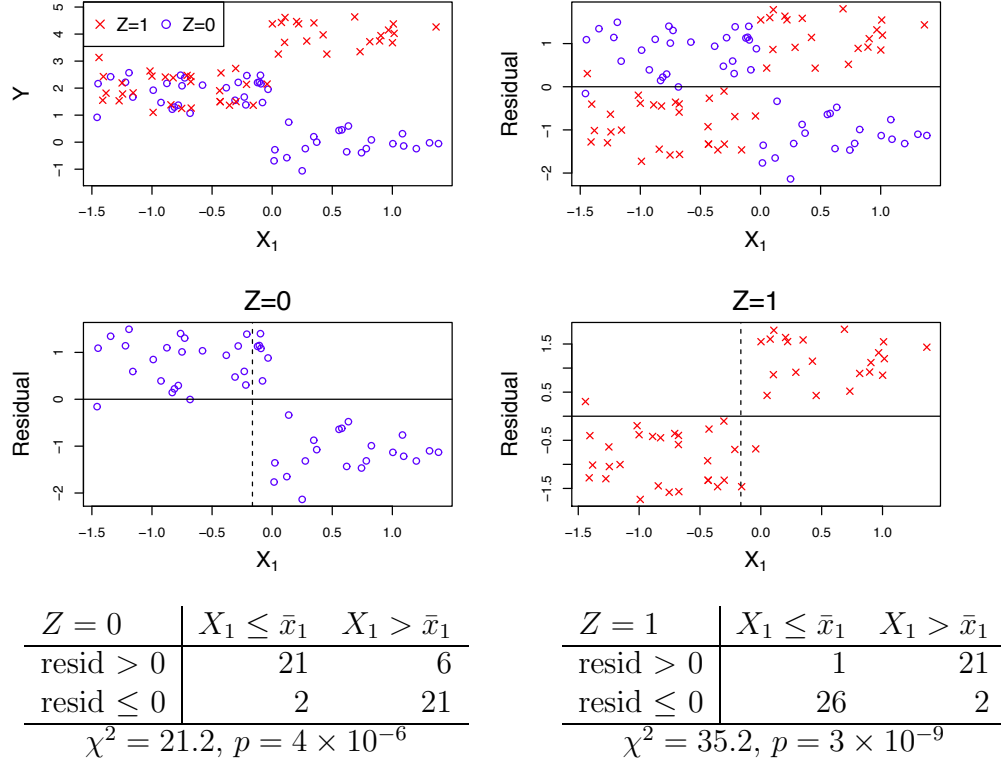


Figure 2: Plots of  $Y$  and residuals vs.  $X_1$  after fitting  $EY = \beta_0 + \beta_1 Z$  to data from model (1); vertical dashed lines indicate sample mean of  $X_1$

The GUIDE linear regression tree algorithm [21] provides an alternative approach that permits  $Y$  and  $Z$  to take more than two values. At each node, we fit a model linear in  $Z$  and select the variable to split it by examining the residual patterns for each level of  $Z$ . Consider, for example, data  $(X_1, X_2, \dots, Y, Z)$  generated from the model

$$Y = 1.9 + 0.2I(Z = 1) - 1.8I(X_1 > 0) + 3.6I(X_1 > 0, Z = 1) + \varepsilon \quad (1)$$

where  $Z = 0, 1$  and  $\varepsilon$  is independent normal. Because the true subgroup is  $X_1 > 0$ , we should split the data using  $X_1$ . Figure 2 shows how this conclusion can be reached from the data alone. The top row plots  $Y$  and the residuals vs.  $X_1$ , where the residuals are from fitting the model  $EY = \beta_0 + \beta_1 Z$ . The middle row plots residuals vs.  $X_1$  for each  $Z$  level. The distinct nonrandom patterns in the latter plots are indicators that  $X_1$  has an interaction with  $Z$ . No other variables can be expected to show such strong patterns. We measure the strength of the interaction by forming a contingency table with the residual signs as rows and grouped values of  $X_1$  (obtained by dividing its values at the sample mean) as shown

In variable selection step, subjects were separated by mean value of a given variable, w/o defining an optimal split value, why this work?

in the bottom row of the figure, computing a chi-squared statistic for each level of  $Z$ , and summing them. Repeating this procedure for each  $X_i$ , we rank the variables and select the one with the largest summed chi-squared to split the data. We call this the Gs method (“s” for sum).

Contingency table tests are convenient because they are quick to compute, can detect a large variety of patterns, and are applicable to categorical  $X$  variables, where we use their values for the columns. Because the latter changes the degrees of freedom (df) of the chi-squared statistics, we need to adjust for differences in df before summing them. We do this by following GUIDE which uses a double application of the Wilson-Hilferty approximation [35] to convert each contingency table chi-squared statistic to a 1-df chi-squared quantile. Specifically, let  $x$  and  $y$  be chi-squared quantiles with  $\nu$  df and  $\mu$ , respectively, degrees of freedom. Then (see [23])

$$y \approx \mu \left[ 1 - 2/(9\mu) + \sqrt{\nu/\mu} \{ (x/\nu)^{1/3} - 1 + 2/(9\nu) \} \right]^3. \quad (2)$$

After a variable is selected, a search is carried out for the best split on the variable that minimizes the sum of squared residuals in the two child nodes and the process is applied recursively to each node. The detailed algorithm, including handling of missing values, is given below.

**Algorithm 1** *Gs split selection.*

1. Fit the least squares model  $EY = \beta_0 + \sum_{z=1}^L \beta_z I(Z = z)$  to the data in the node and compute the residuals. Let  $S_z$  denote the set of observations with  $Z = z$  in the node.
2. For each  $X$  and  $z = 1, 2, \dots, L$ .
  - (a) Form a contingency table from the data in  $S_z$  using the signs (positive vs. non-positive) of the residuals as columns and the (grouped)  $X$  values as rows. If  $X$  is ordinal, divide its values into two groups at the mean. Otherwise, if  $X$  is categorical, let its values define the groups. If there are missing values, add an additional “missing value” group.
  - (b) Compute the chi-squared statistic  $W_z$  for testing independence, and let  $\nu_z$  denote its degrees of freedom. Use (2) to convert  $W_z$  to the 1-df chi-squared quantile

$$r_z(X) = \max \left( 0, \left[ 7/9 + \sqrt{\nu_z} \{ (W_z/\nu_z)^{1/3} - 1 + 2/(9\nu_z) \} \right]^3 \right).$$

3. Treating  $\sum_{z=1}^L r_z(X)$  as a chi-squared variable with  $L$  df, use (2) a second time to convert it to a 1-df chi-squared quantile

$$q(X) = \max \left( 0, \left[ 7/9 + \sqrt{L} \left\{ (L^{-1} \sum_z r_z(X))^{1/3} - 1 + 2/(9L) \right\} \right]^3 \right).$$

4. Let  $X^*$  be the variable with the largest value of  $q(X)$ .



- (a) If  $X^*$  is ordinal, let  $A$  denote the event that  $X^*$  is missing (if any) and  $\bar{A}$  its complement. Then search through the values of  $c$  for the split  $A \cap \{X^* \leq c\}$  or  $\bar{A} \cap \{X^* \leq c\}$  that minimizes the sum of the squared residuals fitted to the two child nodes produced by the split.
- (b) If  $X^*$  is categorical, let  $g$  denote its number of categories (including the missing category, if any). If  $g < 10$ , search over all  $(2^g - 1)$  splits of the form  $X^* \in S$  to find the one that minimizes the sum of squared residuals in the two child nodes. If  $g \geq 10$ , limit the search to  $(g - 1)$  splits as follows.
- i. Label an observation as belonging to class 1 if it has a positive residual and as class 2 otherwise.
  - ii. Order the  $X^*$  values by their proportions of class 1 subjects in the node.
  - iii. Select the split along the ordered  $X^*$  values that yields the greatest reduction in the sum of Gini indices. (This mimics a technique in [6, p. 101] for piecewise constant least-squares regression.)

If there are no missing  $X^*$  values in the training data, future cases missing  $X^*$  are sent to the more populous child node.

Because Gs (as well as MOB) is sensitive to both prognostic and predictive variables, it may be ineffective if only subgroups defined by predictive variables are desired. To see this, suppose now that the data  $(X_1, X_2, \dots, Y, Z)$  are generated from the true model

$$Y = 2I(Z = 1) + I(X_1 > 0) + \varepsilon \quad (3)$$

with  $\varepsilon$  independent normal. The simulated data plots in Figure 3 show that Gs will choose  $X_1$  with high probability even though it is prognostic but not predictive. IT overcomes this by adding the interaction  $I(Z = 1)I(X > c)$  to the fitted model and testing for its significance, but this approach requires searching over the values of  $c$ , which produces selection bias and may be impractical if  $Z$  takes more than two levels. To get around these problems, we instead test for lack of fit of the model

$$EY = \beta_0 + \sum_k \beta_k I(Z = k) + \sum_j \gamma_j I(H = j) \quad (4)$$

where  $H = X$  if it is categorical and is the indicator function  $I(X \leq \bar{x})$  with  $\bar{x}$  being the sample mean of  $X$  at the node otherwise. Then we select the most significant  $X$  to split the data. Turning an ordinal  $X$  into a binary variable may lead to loss of power, but this is compensated by increased sensitivity to interactions of all kinds, including those that cannot be represented by cross-products of indicators. We call this the Gi method (“i” for interaction). The procedure, including handling of missing values, is given next.

**Algorithm 2** *Gi split selection.*

1. For each  $X$  variable at each node:

- (a) If  $X$  is ordinal, divide its values into two groups at its mean. If  $X$  is categorical, let its values define the groups. Add a group for missing values if there are any. Let  $H$  denote the factor variable created from the groups.



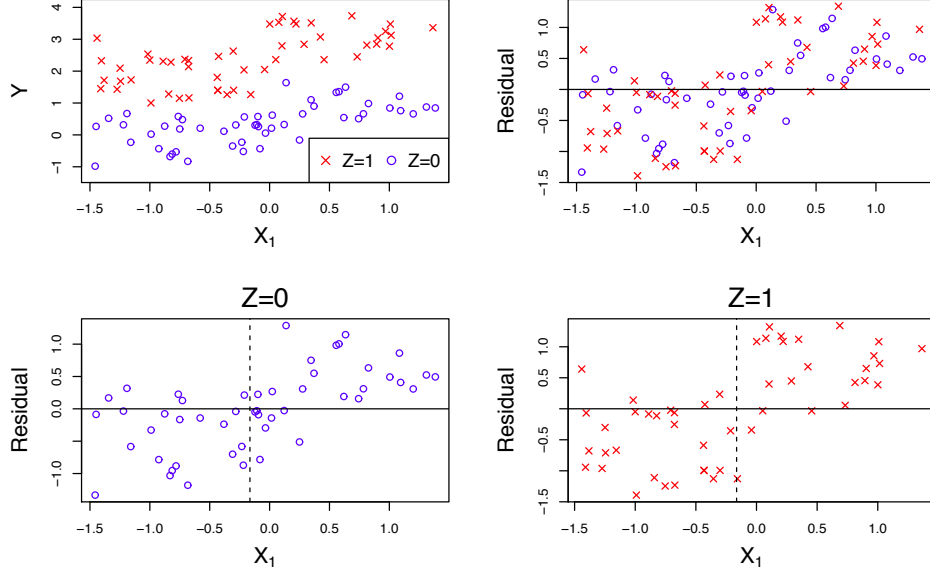


Figure 3: Plots of data and residuals from the model (3) where  $X_1$  is prognostic

- (b) Carry out a **lack-of-fit test** of the model (4) on the data in the node and convert its  $p$ -value to a 1 df chi-squared statistic  $q(X)$ .
2. Let  $X^*$  be the variable with the largest value of  $q(X)$  and use the procedure in Algorithm 1 step 4 to find the split on  $X^*$  that minimizes the sum of squared residuals of the model  $EY = \eta + \sum_k \beta_k I(Z = k)$  fitted to the child nodes.

## 4 Bias and accuracy

### 4.1 Selection bias

It is obviously important for a tree model not to have selection bias if it is used for subgroup identification. At the minimum, this requires that if all the variables are independent of  $Y$ , each  $X_i$  has the same probability of being selected to split each node. We carried out a simulation experiment to compare the methods on this score. The experiment employed two predictors,  $X_1$  and  $X_2$ , and Bernoulli response and treatment variables  $Y$  and  $Z$  each with success probability 0.50. All variables are mutually independent. The distributions of  $X_1$  and  $X_2$  ranged from standard normal, uniform on the integers 1, 2, 3, 4, and equi-probable categorical with 3 and 7 levels, as shown in Table 2.

Based on a sample size of 100 for each iteration, Figure 4 shows the frequency that each method selects  $X_1$  to split the root node over 2500 simulation iterations. Simulation standard errors are less than 0.01. An unbiased method should select  $X_1$  or  $X_2$  with equal probability regardless of their distributions. The results show that IT, QUINT, SIDES and VT have substantial selection biases (QUINT is limited to ordinal  $X_i$ ). IT and QUINT are biased towards selecting the variable that has more splits while SIDES and VT are opposite. In contrast, the selection frequencies of Gs and Gc are all within three simulation standard

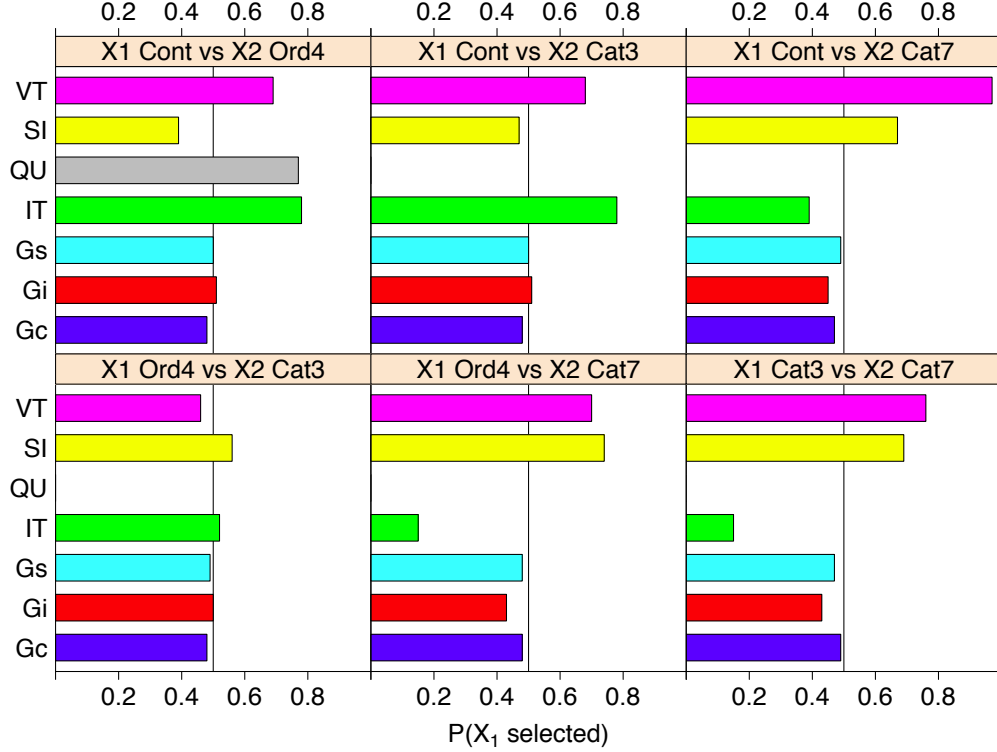


Figure 4: Simulated probabilities that  $X_1$  is selected to split the root node; simulation standard errors less than 0.01. A method is unbiased if it selects  $X_1$  with probability 0.50. SI and QU refer to SIDES and QUINT, respectively. The latter is not applicable to categorical variables.

errors. The frequencies of Gi are also within three standard errors, except when  $X_2$  is categorical with 7 levels where it has a slightly higher chance to be selected.

## 4.2 Accuracy

We use three simulation models to compare the methods in terms of their accuracy in selecting the correct variables and the correct subgroups. Each model employs a binary treatment variable  $Z$  with  $P(Z = 0) = P(Z = 1)$  and 100 variables  $\mathbf{X} = (X_1, X_2, \dots, X_{100})$ , all mutually independent. Each  $X_i$  takes categorical values 0, 1, or 2 (simulating genetic markers with genotypes AA, Aa, and aa), with  $X_1$  and  $X_2$  having identical marginal distribution  $P(X_1 = 0) = 0.4$ ,  $P(X_1 = 1) = 0.465$  and  $P(X_1 = 2) = 0.135$ . The others have marginal distributions  $P(X_j = 0) = (1 - \pi_j)^2$ ,  $P(X_j = 1) = 2\pi_j(1 - \pi_j)$  and  $P(X_j = 2) = \pi_j^2$ , with  $\pi_j$  ( $j = 3, 4, \dots, 100$ ) being independently simulated from a beta distribution with density  $f(x) \propto x(1 - x)^2$ . The models for  $Y$  are:

$$\text{M1: } P(Y = 1|\mathbf{X}) = 0.4 + 0.05I(Z = 1)\{4I(X_1 \neq 0) + 3I(X_2 \neq 0) + I(X_1 \neq 0, X_2 \neq 0)\}$$

$$\text{M2: } P(Y = 1|\mathbf{X}) = 0.3 + 0.2[\{2I(Z = 1) - 1\}I(X_1 \neq 0, X_2 \neq 0) + I(X_3 \neq 0) + I(X_4 \neq 0)]$$

$$\text{M3: } P(Y = 1|\mathbf{X}) = 0.5 + 0.1[2\{I(Z = 1) + I(X_1 \neq 0) + I(X_2 \neq 0)\} - 3].$$

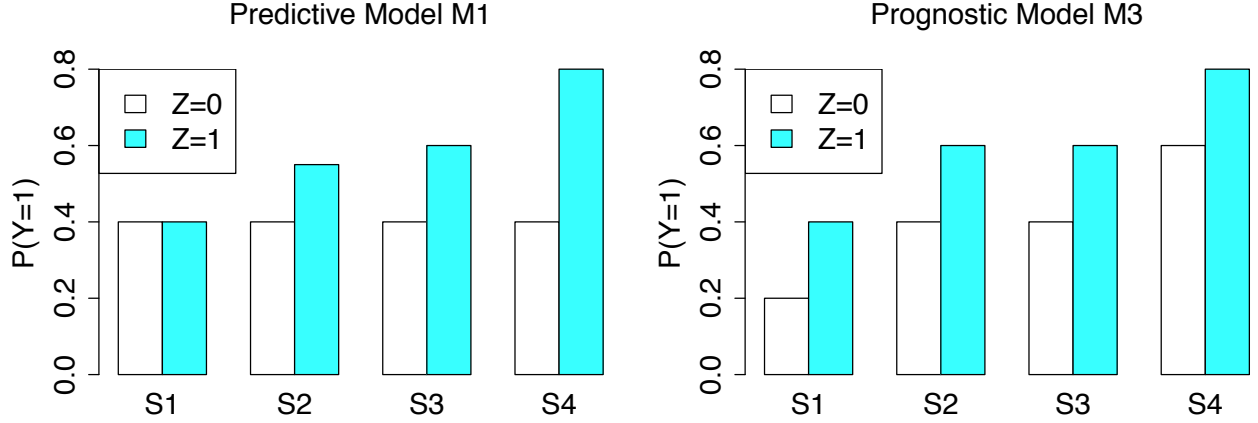


Figure 5: Models M1 and M3. Sets  $S1 = \{X_1 = 0, X_2 = 0\}$ ,  $S2 = \{X_1 = 0, X_2 > 0\}$ ,  $S3 = \{X_1 > 0, X_2 = 0\}$ , and  $S4 = \{X_1 > 0, X_2 > 0\}$ .

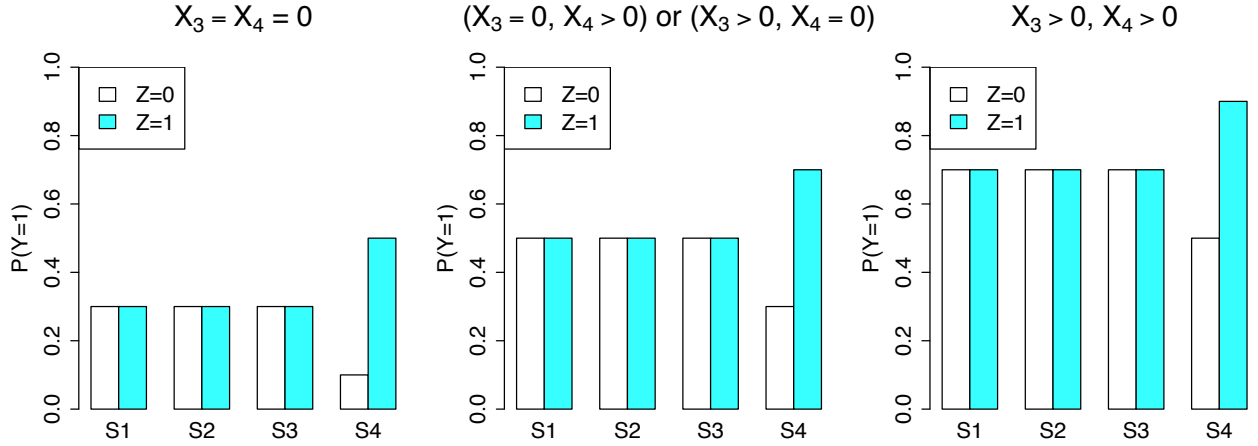


Figure 6: Model M2, with predictive  $X_1$  and  $X_2$  and prognostic  $X_3$  and  $X_4$ . Sets  $S1 = \{X_1 = 0, X_2 = 0\}$ ,  $S2 = \{X_1 = 0, X_2 > 0\}$ ,  $S3 = \{X_1 > 0, X_2 = 0\}$ , and  $S4 = \{X_1 > 0, X_2 > 0\}$ .

Figure 5 shows the values of  $P(Y = 1|\mathbf{X})$  for models M1 and M3. Variables  $X_1$  and  $X_2$  are predictive in M1 but prognostic in M3. Figure 6 shows the values for model M2 which is more complex;  $X_1$  and  $X_2$  are predictive and  $X_3$  and  $X_4$  are prognostic. M2 tests the ability of a method to distinguish between the two variable types.

First we compare the frequencies that  $X_1$  and  $X_2$  are chosen at the first two levels of splits of a tree. For each of 1000 simulation iterations, 100 observations of the vector  $(\mathbf{X}, Y, Z)$  are simulated from each model and a tree is constructed using each method. The frequencies that  $X_1$  or  $X_2$  is selected to split the root node (1st level split) as well as one or both of its child nodes (2nd level split) are graphed in Figure 7. QUINT is excluded because it does not allow categorical variables and the models do not have qualitative interactions. For model M1, where  $X_1$  and  $X_2$  are predictive and there is no other variable associated with  $Y$ , all but SIDES select  $X_1$  or  $X_2$  to split the root node with comparably high frequency. At the second split level, the frequencies for Gs and VT are distinctly higher than those for Gc, Gi and IT, while that for SIDES remains low. Therefore Gs and VT are best and Gc, Gi and IT second best for model M1 on this criterion. The situation is different in M2 which has

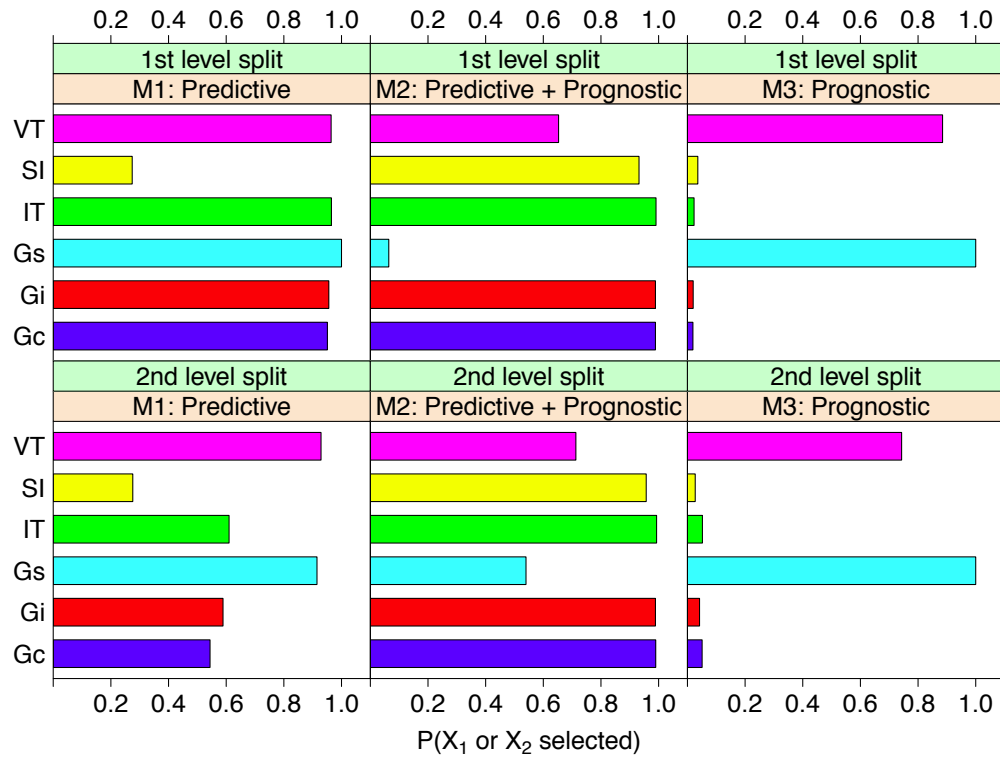


Figure 7: Probabilities that  $X_1$  or  $X_2$  are selected at first and second level splits of trees for models M1, M2, and M3. Long bars are better for M1 and M2, and short bars are better for M3. SI refers to SIDES.

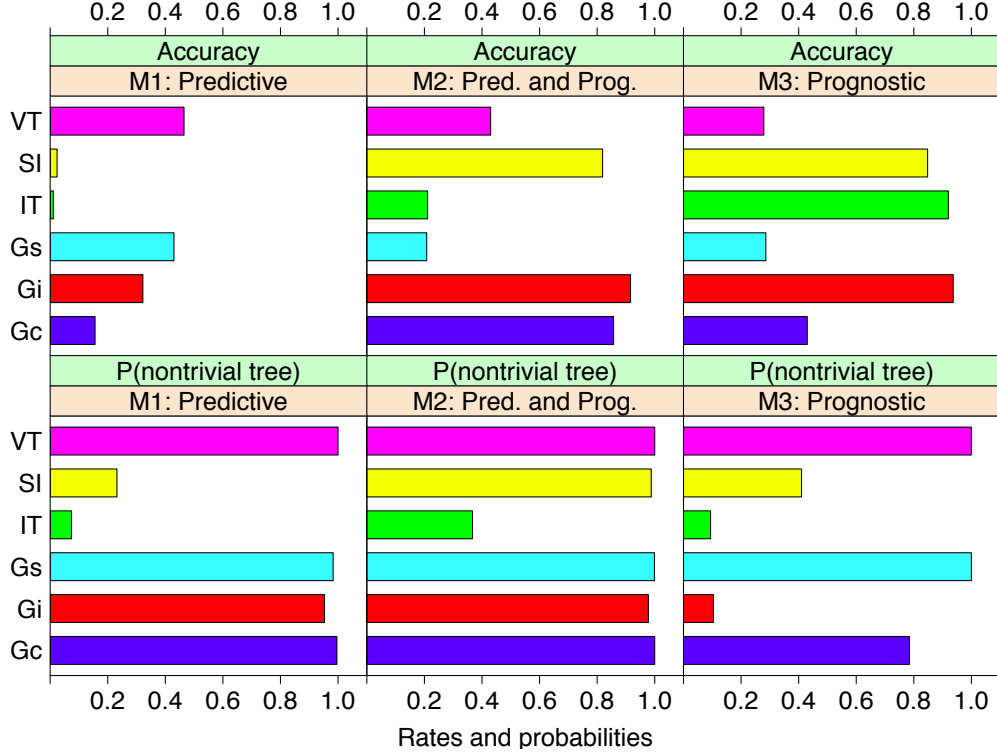


Figure 8: Accuracy rates of subgroup identification and frequencies of nontrivial trees for models M1, M2, and M3. For accuracy, long bars are better. For frequencies of nontrivial trees, long bars are better for M1 and M2 and short bars are better for M3. SI refers to SIDES.

two predictive and two prognostic variables. Now Gc, Gi, IT and SIDES are best and Gs is worst and VT second worst. This shows that Gs and VT have difficulty distinguishing between predictive and prognostic variables. This behavior is repeated in M3 which has no predictive variables. Here the probability that  $X_1$  or  $X_2$  is selected to split the nodes should not be different from that of the other 98 variables, but Gs and VT continue to pick the former with high frequencies. Only Gc, Gi, IT, and SIDES perform correctly in this case.

Next we compare the power of the methods in identifying the correct subgroup. Let  $S$  be any subgroup. Recall from the Introduction that the effect size is  $R(S) = |P(Y = 1|Z = 1, S) - P(Y = 1|Z = 0, S)|$ . The “correct” subgroup  $S^*$  is defined as the maximal (in probability) subgroup  $S$  with the largest value of  $R(S)$ . For models M1 and M2,  $S^* = \{X_1 \neq 0, X_2 \neq 0\}$ ; for M3,  $S^*$  is trivially the whole space because the effect size is 0.20 everywhere.

To estimate accuracy, let  $n(t, y, z)$  denote the number of training samples in node  $t$  with  $Y = y$  and  $Z = z$  and define  $n(t, +, z) = \sum_y n(t, y, z)$  and  $n_t = \sum_z n(t, +, z)$ . Let  $S_t$  be the subgroup defined by  $t$ . The value of  $R(S_t)$  is estimated by  $\hat{R}(S_t) = |n(t, 1, 1)/n(t, +, 1) - n(t, 1, 0)/n(t, +, 0)|$ . The estimate  $\hat{S}$  of  $S^*$  is the subgroup  $S_t$  such that  $\hat{R}(S_t)$  is maximum among all terminal nodes. If  $\hat{S}$  is not unique, take their union. The “accuracy” of  $\hat{S}$  is defined to be  $P(\hat{S})/P(S^*)$  if  $\hat{S} \subset S^*$  and 0 otherwise.

Table 3 and Figure 8 show the estimated accuracies and probabilities of nontrivial trees based on samples of size 100 and 1000 simulation iterations. We see that:

Table 1: Average computational times (sec.) for Model 1

MOB	Gs	Gi	Gc	IT	VT	SI	QU
1.4	4.3	7.0	17.5	130.1	341.1	1601.5	NA

**Model M1.** Gs and VT are best, followed by Gi. Methods IT and SIDES have very low accuracy, due to their high tendency to yield trivial trees and hence no subgroups. The other four methods almost always give nontrivial subgroups.

**Model M2.** Gi has the highest accuracy, at 0.91. It is followed closely by Gc and SIDES at 0.86 and 0.82, respectively. Gs, IT and VT have difficulty distinguishing predictive from prognostic variables. All yield nontrivial trees almost all the time, except for IT which gives a trivial tree 63% of the time.

**Model M3.** Because  $S^*$  is the whole space, the ideal tree should be trivial always. Methods Gi and IT are closest to ideal, yielding trivial trees 90% of the time. In terms of accuracy, Gi, IT, and SIDES are best, having values of 0.94, 0.92, and 0.85, respectively. Gs and VT are the worst, because they produce nontrivial trees all the time.

The above results suggest that Gi is the overall best method in terms of accuracy. It is best in models M2 and M3 and third best in model M1, where it loses to Gs and VT, which are more accurate when there are no prognostic variables. Gc, IT and SIDES are dominated by Gi in every model.

## 5 Bootstrap confidence intervals

Naïve point and interval estimates of the treatment means and differences can certainly be calculated from the training data in each node. Let  $\mu(t, z)$  denote the true mean response for treatment  $z$  in node  $t$  and let  $(y_i, z_i)$ ,  $i = 1, 2, \dots, n_t$ , be the observations in  $t$ . Let  $k_z$  denote the number of observations in  $t$  assigned to treatment  $z$ . Then  $\hat{\mu}(t, z) = k_z^{-1} \sum_{z_i=z} y_i$  is the naïve estimate of  $\mu(t, z)$ ; and if  $\hat{\sigma}(t, z)$  denotes the sample standard deviation of the  $y_i$  among the treatment  $z$  observations in  $t$ , then  $\hat{\mu}(t, z) \pm 2k_z^{-1/2}\hat{\sigma}(t, z)$  is a naïve 95% confidence interval for  $\mu(t, z)$ . Similarly, if  $Z = 0, 1$ , let  $d(t) = \mu(t, 1) - \mu(t, 0)$ . Then  $\hat{d}(t) = \hat{\mu}(t, 1) - \hat{\mu}(t, 0)$  is the naïve estimate of the treatment effect and a naïve confidence interval for  $d(t)$  is the usual two-sample t-interval. Since the nodes in the tree are not fixed in advance but are typically produced by a complex optimization procedure, however, the validity of these estimates should not be taken for granted. For example, SIDES employs adjustments to the naïve p-values of treatment effects in the nodes to control bias.

To see the extent of the bias for Gi and Gs, we carried out a simulation experiment using models M1 and M2. The experimental design is an  $r$ -replicate ( $r = 2, 4$ ) of a  $3^4$  factorial in variables  $X_1, X_2, X_3, X_4$ , each taking values 0, 1, and 2, with  $Z$  independent Bernoulli with probability 0.50, and the binary response  $Y$  simulated according to models M1 or M2. In each simulation trial, a Gi or Gs tree  $T$  is constructed from the training data. If  $T$  is nontrivial, we record the average values of  $\hat{\mu}(t, z) - \mu(t, z)$  and  $\hat{d}(t) - d(t)$  over the nodes  $t$  and the proportions of times each naïve confidence interval contains the true estimand.

Columns 3–8 of Table 4 show the estimated bias and coverage probabilities of the intervals over 2000 simulation trials with nontrivial trees. The biases are remarkably small (the true means range from 0.30 to 0.90). We attribute this to Gi and Gs being not directed at finding splits to maximize or minimize the treatment effect, unlike SIDES and QUINT. The coverage probabilities, on the other hand, are all too low, although there is a perceptible improvement as  $R$  increases..

We use the following method to construct better intervals by using the bootstrap to estimate the standard deviations of the naïve estimates. Let  $\mathcal{L}$  denote a given data set and let  $T$  denote the regression tree constructed from it. Let  $\mathcal{L}_j^*$  ( $j = 1, 2, \dots, J$ ) be a bootstrap training sample from  $\mathcal{L}$  and let  $T_j^*$  be the tree constructed from  $\mathcal{L}_j^*$  with naïve estimates  $\hat{\mu}_j^*(t^*, z)$  for terminal nodes  $t^*$  in  $T_j^*$ . Let  $n_z(t \cap t^*)$  be the number of treatment  $z$  observations from  $\mathcal{L}$  that belong to  $t \cap t^*$  and define

$$\bar{\mu}_j^*(t, z) = \sum_{t^*} n_z(t \cap t^*) \hat{\mu}_j^*(t^*, z) / \sum_{t^*} n_z(t \cap t^*).$$

The bootstrap estimate of the variance of  $\hat{\mu}(t, z)$  is the sample variance  $s_\mu^2(t, z)$  of  $\{\bar{\mu}_1^*(t, z), \bar{\mu}_2^*(t, z), \dots, \bar{\mu}_J^*(t, z)\}$  and a 95% bootstrap confidence interval for  $\mu(t, z)$  is  $\hat{\mu}(t, z) \pm 2s_\mu(t, z)$ . If  $Z$  takes values 0 and 1, let  $\bar{d}_j^*(t) = \bar{\mu}_j^*(t, 1) - \bar{\mu}_j^*(t, 0)$ . Then a 95% confidence interval for  $d(t)$  is  $\hat{d}(t) \pm 2s_d(t)$  where  $s_d^2(t)$  is the sample variance of  $\{\bar{d}_1^*(t), \bar{d}_2^*(t), \dots, \bar{d}_J^*(t)\}$ .

The rightmost three columns of Table 4 give the simulated coverage probabilities of the bootstrap intervals using  $J = 100$ . There is a clear improvement over the naïve intervals. In particular, the bootstrap intervals for the treatment effect  $d(t)$  are remarkably accurate across the two models and two methods. The worst performance occurs in model M1 for  $Z = 0$ , where the true treatment mean is 0.40 in all nodes (see Figure 5).

## 6 Censored data

Several obstacles stand in the way of direct extension of Gi and Gs to data with censored response variables. The obvious approach of replacing least squares fits with proportional hazards models in the nodes [26, 33] is problematic because Gi and Gs employ chi-squared tests on residuals and their signs. Although there are many definitions of such residuals [3], it is unclear if any will serve the purpose here. Besides, as noted earlier, fitting a separate proportional hazards model in each node yields different baseline cumulative hazard functions. As a result, the whole model no longer has proportional hazards and hence regression coefficients between nodes cannot be compared. To preserve this property, a common estimated baseline cumulative hazard function is required. We solve these problems with the old trick of using Poisson regression to fit proportional hazards models.

Let  $u_i$  and  $\mathbf{x}_i$  denote the survival time and covariate vector of subject  $i$ . Let  $s_i$  be an independent observation from some censoring distribution and let  $\delta_i = I(u_i < s_i)$  be the event indicator. The observed data vector corresponding to subject  $i$  is  $(y_i, \delta_i, \mathbf{x}_i)$ , where  $y_i = \min(u_i, s_i)$ ,  $i = 1, 2, \dots, n$ . Let  $F(u, \mathbf{x})$  and  $\lambda(u, \mathbf{x})$  denote the distribution and hazard functions, respectively, at  $\mathbf{x}$ . The proportional hazards model specifies that  $\lambda(u, \mathbf{x}) = \lambda_0(u) \exp(\eta)$ , where  $\lambda_0(u)$  is the baseline hazard and  $\eta = \beta' \mathbf{x}$  is a linear function of the covariates. Let  $\Lambda(u, \mathbf{x}) = \int_{-\infty}^u \lambda(z, \mathbf{x}) dz$  denote the cumulative hazard function



and let  $\Lambda_0(u) = \Lambda(u, \mathbf{0})$  be the baseline cumulative hazard. Then the density function is  $f(u, \mathbf{x}) = \lambda_0(u) \exp\{\eta - \Lambda_0(u) \exp(\eta)\}$ . Letting  $\mu_i = \Lambda_0(y_i) \exp(\eta_i)$ , the loglikelihood can be expressed as

$$\begin{aligned} & \sum_{i=1}^n \delta_i \log f(y_i, \mathbf{x}_i) + \sum_{i=1}^n (1 - \delta_i) \log\{1 - F(y_i, \mathbf{x}_i)\} \\ &= \sum_{i=1}^n [\delta_i \{\log \Lambda_0(y_i) + \eta_i\} - \Lambda_0(y_i) \exp(\eta_i) + \delta_i \log\{\lambda_0(y_i)/\Lambda_0(y_i)\}] \\ &= \sum_{i=1}^n (\delta_i \log \mu_i - \mu_i) + \sum_{i=1}^n \delta_i \log\{\lambda_0(y_i)/\Lambda_0(y_i)\}. \end{aligned}$$

The first term on the right is the kernel of the loglikelihood for  $n$  independent Poisson variables  $\delta_i$  with means  $\mu_i$  and the second term is independent of the covariates (see, e.g., [4, 18]). If the  $\Lambda_0(y_i)$  values are known, the vector  $\beta$  may be estimated by treating the event indicators  $\delta_i$  as independent Poisson variables distributed with means  $\Lambda_0(y_i) \exp(\beta' \mathbf{x}_i)$ .

Thus we can construct a proportional hazards regression tree by iteratively fitting a Poisson regression tree [8, 22], using  $\delta_i$  as Poisson responses, the treatment indicators as predictor variables, and  $\log \Lambda_0(y_i)$  as offset variable. Gi and Gs employ loglinear model goodness-of-fit tests [2, p.212] to the fitted values to obtain the split variables and split points. At the first iteration,  $\Lambda_0(y_i)$  is estimated by the Nelson-Aalen [1, 7] method. After each iteration, the estimated relative risks of the observations from the tree model are used to update  $\Lambda_0(y_i)$  for the next iteration (see, e.g., [19, p. 361]). The results reported here are obtained with five iterations.

Figure 1 gives the results of applying these techniques to the breast cancer data. Gi and Gs each splits the data once, at `progrec`  $\leq 21$  and `pnodes`  $\leq 3$ , respectively. The corresponding Kaplan-Meier curves in the figure show that `progrec` is predictive and `pnodes` is prognostic. The 95% bootstrap confidence intervals of  $\exp(\beta)$ , the relative risk of hormone therapy versus no therapy, are shown beside the terminal nodes of the trees. They are constructed as for uncensored response data, with the regression coefficient  $\beta$  replacing the mean response. Specifically, let  $\mathcal{L}$  and  $T$  denote the training sample and the tree constructed from it. Let  $\mathcal{L}_j^*$  and  $T_j^*$  denote the corresponding  $j$ th bootstrap sample and tree, for  $j = 1, 2, \dots, J$ . Let  $\hat{\beta}(t)$  and  $\hat{\beta}_j^*(t^*)$  denote the estimates of  $\beta$  in nodes  $t \in T$  and  $t^* \in T_j^*$  based on  $\mathcal{L}$  and  $\mathcal{L}_j^*$ , respectively, and let  $n(A)$  denote the number of cases in  $\mathcal{L}$  that belong to any set  $A$ . Define  $\bar{\beta}_j^*(t) = \sum_{t^*} n(t \cap t^*) \hat{\beta}_j^*(t^*) / \sum_{t^*} n(t \cap t^*)$ . The bootstrap estimate of the variance of  $\hat{\beta}(t)$  is the sample variance  $s_{\beta}^2(t)$  of  $\{\bar{\beta}_1^*(t), \bar{\beta}_2^*(t), \dots, \bar{\beta}_J^*(t)\}$  and a 95% bootstrap confidence interval for  $\beta(t)$  is  $\hat{\beta}(t) \pm 2s_{\beta}(t)$ .

## 7 Importance scoring and thresholding

When there are many variables, it may be useful or necessary to reduce their number by some form of variable selection. One way to accomplish this is to rank them in their order of importance and select a top-ranked subset. Lack of a proper definition of “importance”

has led to many scoring methods being proposed. Few methods include thresholds for identifying the noise variables. For example, CART and random forest use the information from surrogate splits to compute scores but not thresholds.

Following [24], **we score the importance of a variable  $X$  in terms of the 1-df chi-squared statistics computed during variable selection.** Specifically, let  $q_t(X)$  be the value of  $q(X)$  (see Algorithms 1 and 2) at node  $t$  and  $n_t$  be the number of observations in  $t$ . We define the importance score of  $X$  to be  $\text{Imp}(X) = \sum_t n_t q_t(X)$  and approximate its null distribution with a scaled chi-squared using the Satterthwaite method [28]. This procedure is similar to that in [24] except for two differences. First, the latter employs the weight  $\sqrt{n_t}$  is used instead of  $n_t$  in the definition of  $\text{Imp}(X)$ . The new definition increases the probability that the variable selected to split the root node is top-ranked. The other change is in the choice of threshold. In [24], the threshold is the  $K^{-1}(K-1)$ -quantile of the approximating distribution of  $\text{Imp}(X)$ , where  $K$  is the number of predictor variables, the motivation being that  $1/K$  of the unimportant variable are expected to be found important. It is difficult to compute the  $K^{-1}(K-1)$ -quantile of the distribution, however, if  $K$  is large. Therefore the threshold is defined to be the 0.95-quantile instead. With this definition, Gi identifies only **progrec** as important. The important variables for Gs are **pnodes**, **progrec**, and **estrec**, in descending order.

## 8 Application to data with missing values

Missing values pose two problems for tree construction. The first is how to deal with them during split selection and the second is how to send observations with missing values through a split. CART uses a system of surrogate splits that is biased toward choosing variables with few missing values [17, 32]. For variable selection, Gc, Gi and Gs create a “missing” category in each contingency table. Each split has the form  $x \in S$ , where the set  $S$  may contain the missing values. There is some evidence that this technique is best among classification tree methods if the response variable takes two values [10].

We illustrate the method on a real data set from a retrospective candidate gene study. Owing to confidentiality reasons, the data and solutions are described in general terms here. A total of 1504 subjects were randomized to treatment or placebo and the response is survival time in days, with sixty-three percent censored. The explanatory variables consist of 17 continuous-valued baseline measures (**a1**, **a2**, and **b01–b15**) and 288 categorical variables, of which 6 are baseline measures (**c0–c5**) and the rest are genetic variables (**g001–g282**), each with two or three levels. More than 95% (1435/1504) of the subjects have missing values in one or more explanatory variables and only 7 variables (**a1**, **a2**, **b3**, **c0**, **c4**, **b15**, **g272**) are completely observed.

Although the treatment effect is statistically significant, its magnitude is small. The question is whether there are subgroups for which there are larger treatment effects. Owing to the large number of variables, a traditional Cox proportional hazards model is inapplicable without some sort of variable selection, even if restricted to the subset of complete observations.

The Gs model, shown in Figure 9, splits only once, on **a2**. If the latter is less than 0.1 or missing, there is little difference in survival probability between treated and untreated, as

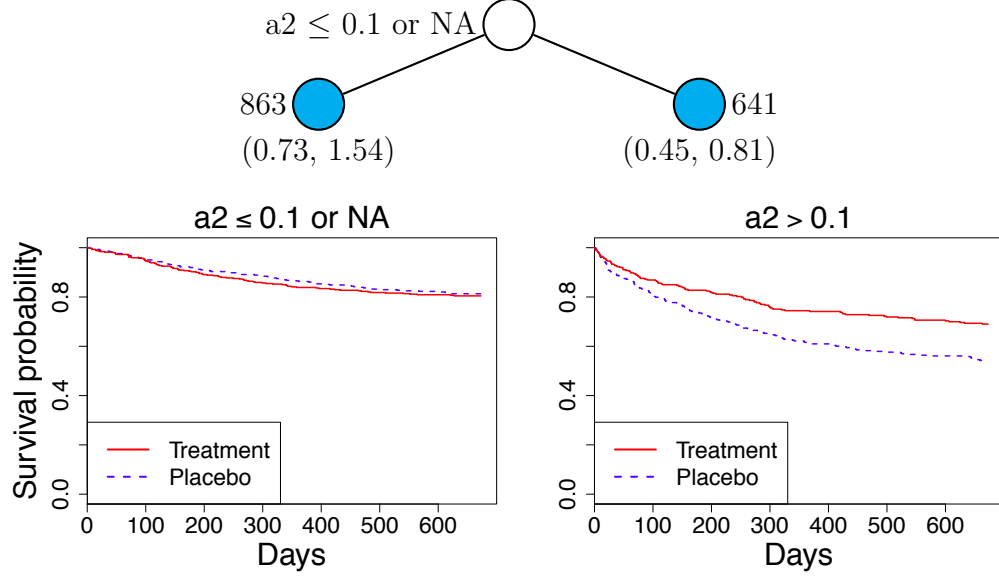


Figure 9: Gs model for gene data. At each node, a case goes to the left child node if and only if the stated condition is satisfied. Sample sizes are beside terminal nodes and 95% bootstrap intervals for relative risk of recurrence for treatment versus placebo are below the nodes.

shown by the Kaplan-Meier curves below the node. Otherwise, the difference is statistically significant: a 95% bootstrap confidence interval (based on 100 bootstrap iterations) for relative risk (treatment vs. placebo) is (0.45, 0.81). The importance scoring method identifies 27 and 28 important variables for Gi and Gs, respectively. The trees constructed from these variables, however, are unchanged.

## 9 Conclusion

Regression trees are natural for subgroup identification because they find subgroups that are interpretable. But interpretability is advantageous only if the algorithms that produce the trees do not possess selection bias. We have introduced three algorithms that are practically unbiased in this respect. Gc is simplest because it can be implemented with any classification tree algorithm (preferably one without selection bias) by appropriate definition of a class variable. It is limited, however, to binary-value response and treatment variables. Further, some modification of the classification tree algorithm may be needed to disallow splits that yield nodes with pure classes.

Gs is a descendant of the GUIDE regression tree algorithm. As a result, it is more general than Gc, being applicable to any kind of ordinal response variables, including those subject to censoring, to multi-valued treatment variables, and to all types of predictors variables, with missing values or not. If there is no censoring, Gs borrows all the ingredients of GUIDE. The main differences lie in the use of the treatment variable as the sole predictor in a linear model fitted to each node, the construction of a separate chi-squared test of the residuals versus each predictor for each treatment level, and the sum of the Wilson-Hilferty

transformed chi-squared statistics to form a single criterion for split variable selection at each node. As the example in Figure 3 demonstrates, however, Gs can be distracted by the presence of prognostic variables.

Gi is our preferred solution if the goal is to find subgroups defined by predictive variables only. To avoid being distracted by prognostic variables, Gi uses a chi-squared test of treatment-covariate interaction to select a split variable at each node. It is therefore similar in spirit to the IT method. But unlike the latter, which searches for the split variable and the split point at the same time, Gi uses the chi-squared test for variable selection only. Besides avoiding selection bias, this approach yields the additional benefit of reduced computation time.

We extend Gi and Gs to censored time-to-event data by fitting a tree-structured proportional hazards model to the data by means of Poisson regression. Poisson residuals are easier to use for our purposes than those from proportional hazards models. Further, this technique gives a common baseline cumulative hazard function and allows comparisons of estimates between nodes. The price is increased computing time due to the need for iterative updates of the estimated baseline cumulative hazard function, but the expense is not large relative to the other methods, as shown by the average computing times to construct one tree for model M1 in Table 5. (Model M1 has only  $X$  variables that take three values each; the relative speeds of Gc, Gi and Gs will be greater if these variables take more values.)

Subgroup identification is prone to error if the number of predictor variables greatly exceeds the sample size, because the chance of finding the correct variables can be small, as our simulation results demonstrate. If the number of variables is large, it is often helpful to eliminate some of the irrelevant variables with importance score thresholds and then construct the trees with the remaining ones. Our scoring and thresholding method is particularly convenient for this purpose because, unlike other approaches, it does not require data resampling and hence is much quicker.

To our knowledge, there has not been an effective method of confidence interval construction for the estimates in the nodes of a regression tree. The main difficulty is the numerous levels of selection typically involved in tree construction. Not surprisingly, naïve intervals that ignore the variability due to selection are overly optimistic. To solve this problem, we have to account for this extra variability. We do this by using a bootstrap method to estimate the true standard errors of the estimated treatment effects. Because each bootstrapped tree is likely different (and different from the original), we do not obtain an interval for each of its nodes. Instead, we average the bootstrap treatment effects within each node of the original tree and use the averages to estimate the standard errors of the original treatment effects. We do not yet have theoretical proof of the consistency of this procedure, but the empirical results are promising.

The Gi and Gs methods are implemented in the GUIDE computer program which can be obtained from [www.stat.wisc.edu/~loh/guide.html](http://www.stat.wisc.edu/~loh/guide.html).

## Acknowledgments

We are very grateful to Xiaogang Su, Jared Foster, Jue Hou, and Elise Dusseldorp for sharing with us their R programs for IT, VT, SIDES, and QUINT, respectively, and for

their patience in answering our numerous questions. We also thank Lei Shen for helpful comments on the manuscript. This work was partially supported by U.S. Army Research Office grant W911NF-09-1-0205, NSF grant DMS-1305725, NIH grant P50CA143188, and a grant from Eli Lilly and Company.

## References

- [1] Aalen, O. O. (1978). Nonparametric inference for a family of counting processes. *Annals of Statistics*, 6:701–726.
- [2] Agresti, A. (2007). *An Introduction to Categorical Data Analysis*. Wiley, 2nd edition.
- [3] Ahn, H. and Loh, W.-Y. (1994). Tree-structured proportional hazards regression modeling. *Biometrics*, 50:471–485.
- [4] Aitkin, M. and Clayton, D. (1980). The fitting of exponential, Weibull and extreme value distributions to complex censored survival data using GLIM. *Applied Statistics*, 29:156–163.
- [5] Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.
- [6] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, California.
- [7] Breslow, N. (1972). Contribution to the discussion of regression models and life tables by D. R. Cox. *Journal of the Royal Statistical Society, Ser. B*, 34:216–217.
- [8] Chaudhuri, P., Lo, W.-D., Loh, W.-Y., and Yang, C.-C. (1995). Generalized regression trees. *Statistica Sinica*, 5:641–666.
- [9] Ciampi, A., Negassa, A., and Lou, Z. (1995). Tree-structured prediction for censored survival data and the Cox model. *Journal of Clinical Epidemiology*, 48:675–689.
- [10] Ding, Y. and Simonoff, J. S. (2010). An investigation of missing data methods for classification trees applied to binary response data. *Journal of Machine Learning Research*, 11:131–170.
- [11] Dusseldorp, E. and Van Mechelen, I. (2013). Qualitative interaction trees: a tool to identify qualitative treatment-subgroup interactions. *Statistics in Medicine*. In press.
- [12] Everitt, B. S. and Hothorn, T. (2006). *A Handbook of Statistical Analyses Using R*. Chapman and Hall/CRC.
- [13] Foster, J. C., Taylor, J. M. G., and Ruberg, S. J. (2011). Subgroup identification from randomized clinical trial data. *Statistics in Medicine*, 30:2867–2880.
- [14] Hothorn, T., Hornik, K., Strobl, C., and Zeileis, A. (2012). *Party: a laboratory for recursive partytioning*. R package version 1.0-1.

- [15] Italiano, A. (2011). Prognostic or predictive? It’s time to get back to definitions! *Journal of Clinical Oncology*, 29:4718.
- [16] Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29:119–127.
- [17] Kim, H. and Loh, W.-Y. (2001). Classification trees with unbiased multiway splits. *Journal of the American Statistical Association*, 96:589–604.
- [18] Laird, N. and Olivier, D. (1981). Covariance analysis of censored survival data using log-linear analysis techniques. *Journal of the American Statistical Association*, 76:231–240.
- [19] Lawless, J. F. (1982). *Statistical Models and Methods for Lifetime Data*. Wiley, New York.
- [20] Lipkovich, I., Dmitrienko, A., Denne, J., and Enas, G. (2011). Subgroup identification based on differential effect search — a recursive partitioning method for establishing response to treatment in patient subpopulations. *Statistics in Medicine*, 30:2601–2621.
- [21] Loh, W.-Y. (2002). Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, 12:361–386.
- [22] Loh, W.-Y. (2006). Regression tree models for designed experiments. In Rojo, J., editor, *Second E. L. Lehmann Symposium*, volume 49, pages 210–228. Institute of Mathematical Statistics Lecture Notes-Monograph Series.
- [23] Loh, W.-Y. (2009). Improving the precision of classification trees. *Annals of Applied Statistics*, 3:1710–1737.
- [24] Loh, W.-Y. (2012). Variable selection for classification and regression in large  $p$ , small  $n$  problems. In Barbour, A., Chan, H. P., and Siegmund, D., editors, *Probability Approximations and Beyond*, volume 205 of *Lecture Notes in Statistics—Proceedings*, pages 133–157, New York. Springer.
- [25] Loh, W.-Y. and Shih, Y.-S. (1997). Split selection methods for classification trees. *Statistica Sinica*, 7:815–840.
- [26] Negassa, A., Ciampi, A., Abrahamowicz, M., Shapiro, S., and Boivin, J. R. (2005). Tree-structured subgroup analysis for censored survival data: validation of computationally inexpensive model selection criteria. *Statistics and Computing*, 15:231–239.
- [27] Peters, A. and Hothorn, T. (2012). Improved predictors. R package version 0.8-13.
- [28] Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components. *Biometrics Bulletin*, 2:110–114.
- [29] Sauerbrei, W. and Royston, P. (1999). Building multivariable prognostic and diagnostic models: transformation of the predictors by using fractional polynomials. *Journal of the Royal Statistical Society, Series A*, 162:71–94.

- [30] Schmoor, C., Olschewski, M., and Schumacher, M. (1996). Randomized and non-randomized patients in clinical trials: experiences with comprehensive cohort studies. *Statistics in Medicine*, 15:263–271.
- [31] Schumacher, M., Baster, G., Bojar, H., Hübner, K., Olschewski, M., Sauerbrei, W., Schmoor, C., Beyerle, C., Newmann, R. L. A., and Rauschecker, H. F. (1994). Randomized  $2 \times 2$  trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. *Journal of Clinical Oncology*, 12:2086–2093.
- [32] Strobl, C., Boulesteix, A.-L., Zeileis, A., and Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8:25.
- [33] Su, X., Zhou, T., Yan, X., Fan, J., and Yang, S. (2008). Interaction trees with censored survival data. *International Journal of Biostatistics*, 4. Article 2.
- [34] Therneau, T. M. and Atkinson, B. (2012). *Rpart: Recursive partitioning*. R package version 3.1-51.
- [35] Wilson, E. B. and Hilferty, M. M. (1931). The distribution of chi-square. *Proceedings of the National Academy of Sciences of the United States of America*, 17:684–688.



Table 2: Four types of distributions of  $X_1$  and  $X_2$ .

Notation	Type	Distribution
Cont	Continuous	Standard normal
Ord4	Ordinal	Discrete uniform with 4 levels
Cat3	Categorical	Discrete uniform with 3 levels
Cat7	Categorical	Discrete uniform with 7 levels

Table 3: Accuracy rates of subgroup selection and frequencies of nontrivial trees. Larger values are better.

Model	Type	Gi	Gs	Gc	IT	SIDES	VT
M1	Accuracy	0.322	0.430	0.150	0.011	0.024	0.465
M1	P(nontrivial tree)	0.953	0.983	0.990	0.074	0.232	1.000
M2	Accuracy	0.913	0.204	0.855	0.211	0.819	0.430
M2	P(nontrivial tree)	0.979	0.999	1.000	0.367	0.988	1.000
M3	Accuracy	0.939	0.285	0.519	0.920	0.848	0.279
M3	P(nontrivial tree)	0.104	1.000	0.693	0.094	0.410	1.000

Table 4: Bias of estimated treatment means and their difference (averaged over nodes of each tree) and coverage probabilities of naïve and bootstrap 95% intervals, based on 1000 simulations trials of nontrivial trees, with 100 bootstrap iterations per trial.  $r$  is the number of replicates of a  $3^4$  design.

$r$	Expt	Bias of naïve estimates of means and difference			Coverage probabilities of 95%					
		$\mu(t, 0)$	$\mu(t, 1)$	$d(t)$	naïve intervals			bootstrap intervals		
		$\mu(t, 0)$	$\mu(t, 1)$	$d(t)$	$\mu(t, 0)$	$\mu(t, 1)$	$d(t)$	$\mu(t, 0)$	$\mu(t, 1)$	$d(t)$
2	M1-Gi	6.8E-3	-2.2E-2	-2.9E-2	0.821	0.811	0.818	0.892	0.955	0.934
	M1-Gs	4.4E-3	-1.9E-2	-2.3E-2	0.819	0.800	0.857	0.907	0.952	0.935
	M2-Gi	6.8E-3	-2.0E-2	-2.6E-2	0.835	0.846	0.836	0.937	0.947	0.941
	M2-Gs	3.5E-4	-1.5E-2	-1.5E-2	0.871	0.861	0.907	0.953	0.965	0.942
4	M1-Gi	3.0E-2	-1.6E-2	-1.9E-2	0.880	0.874	0.889	0.903	0.972	0.957
	M1-Gs	3.7E-3	-1.5E-2	-1.9E-2	0.869	0.862	0.888	0.916	0.967	0.955
	M2-Gi	1.1E-3	-7.5E-3	-8.6E-3	0.896	0.915	0.911	0.966	0.967	0.963
	M2-Gs	-3.8E-3	-9.4E-3	-5.7E-3	0.888	0.913	0.916	0.968	0.973	0.950

Table 5: Average times (sec.), over 500 simulation trials of model M1, to construct one tree on a 2.66GHz Intel processor; QUINT does not allow categorical variables

Gs	Gi	Gc	IT	VT	SIDES	QUINT
4.3	7.0	17.5	130.1	341.1	1601.5	NA