Statistics
in Medicine

Research Article

Received 11 February 2014, Accepted 20 January 2015 Published online 5 February 2015 in Wiley Online Library

(wileyonlinelibrary.com) DOI: 10.1002/sim.6454

A regression tree approach to identifying subgroups with differential treatment effects

Wei-Yin Loh,[a*†] Xu He[b] and Michael Man[c]

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 data from randomized trials with two or more treatments, censored response variables, and missing values in the predictor variables. The algorithms extend the generalized unbiased interaction detection and estimation (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. A bootstrap technique is used to construct confidence intervals for the treatment effects in each node. The methods are compared using real and simulated data. Copyright © 2015 John Wiley & Sons, Ltd.

Keywords: missing values; proportional hazards; selection bias; bootstrap

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 is ineffective. This problem of searching for subgroups with differential treatment effects is known as *subgroup identification* [1–3].

To fix ideas, suppose that the response variable Y is uncensored and the treatment variable Z takes values $l = 1, 2, \ldots, L$. 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 [4, 5] with 686 subjects where the response is recurrence-free survival time in days. The trial was designed as a 2×2 factorial comparing three versus six cycles of chemotherapy and presence versus absence of hormone therapy, but the data contain no information on the number of cycles, presumably because it was previously found not significant [6]. Median follow-up time was nearly 5 years, and 387 subjects did not experience a recurrence of the disease during the trial (54% 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]*Department of Statistics, University of Wisconsin, Madison, WI 53706, U.S.A.*
[b]*Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China*
[c]*Eli Lilly and Company, Indianapolis, IN 46285, U.S.A.*
[*]*Correspondence to: Wei-Yin Loh, Department of Statistics, University of Wisconsin, Madison, WI 53706, U.S.A.*
[†]*E-mail: loh@stat.wisc.edu*

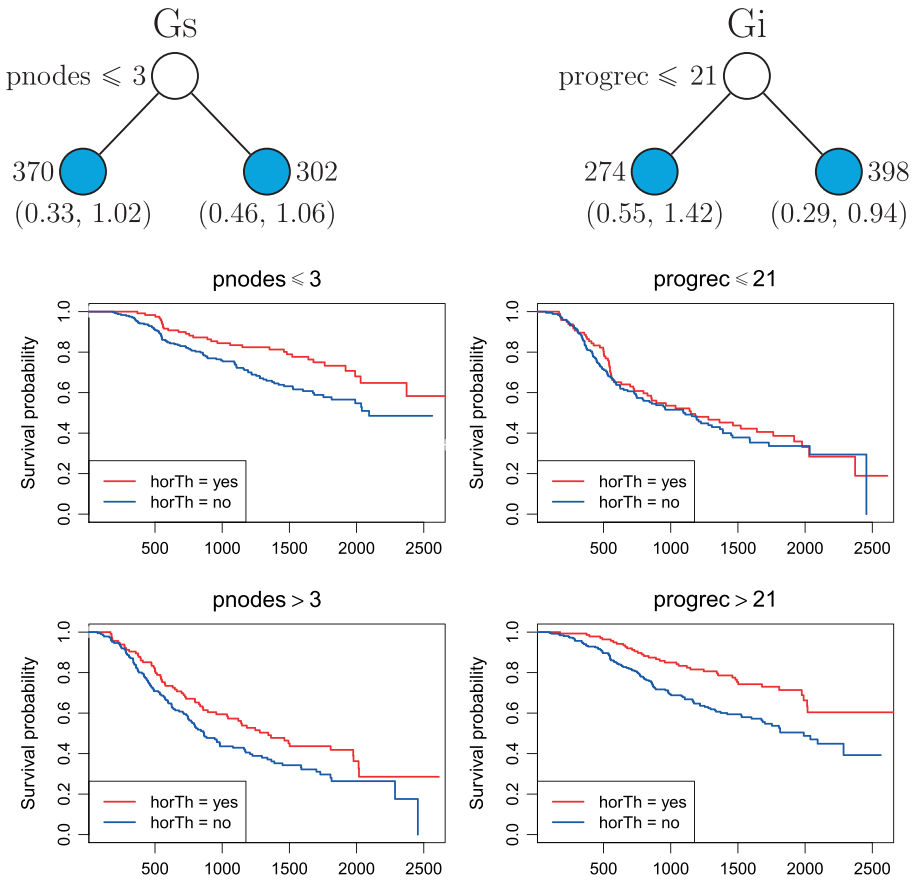


Figure 1. Gs (left) and Gi (right) tree models and Kaplan–Meier curves for breast cancer data. At each intermediate node, an observation goes to the left child node if and only if the displayed condition is satisfied. Sample sizes are beside terminal nodes; 95% bootstrap confidence intervals of relative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

A standard proportional hazards regression model finds hormone therapy to have a significant positive effect on survival, with and without adjusting for the covariates [7, 8]. It is interesting, however, to find out if a subgroup exists where the treatment has no 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 genetic data, these models cannot be used without prior variable selection. Regression tree models are good 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 current medical literature [9, 10], 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 Gs model on the left, variable `pnodes` is prognostic: recurrence probability is reduced if `pnodes` >3, with and without treatment. In the Gi model on the right, variable `progrec` is predictive: hormone therapy has little effect if `progrec` $\leqslant 21$ 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 Section 2. Section 3 presents the new algorithms for uncensored response variables. Section 4 compares the selection bias and accuracy of the new and old methods, and Section 5 proposes a bootstrap technique for computing confidence intervals of treatment and other effects in the subgroups. Section 6 extends the algorithms to censored survival data, and Section 7 obtains importance scores for ranking the variables and thresholds for identifying the

unimportant ones. Section 8 gives an application to a retrospective candidate gene study where there are large numbers of missing values, and Section 9 concludes the article with some closing remarks. It is assumed throughout that the data are obtained from randomized experiments (see, e.g., [11] for regression tree models for observational data).

2. Previous work

Let u_i and $\mathbf{x}_i$ denote the (actual but possibly unobserved) 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 of subject i is $(y_i, \delta_i, \mathbf{x}_i)$, where $y_i = \min(u_i, s_i)$, $i = 1, 2, \ldots, n$. Let $\lambda(y, \mathbf{x})$ denote the hazard function at $\mathbf{x} = (x_1, x_2, \ldots, x_M)$. The proportional hazards model specifies that $\lambda(y, \mathbf{x}) = \lambda_0(y)\exp(\eta)$, where $\lambda_0(y)$ is a baseline hazard and $\eta = \boldsymbol{\beta}'\mathbf{x}$ is a linear function of the covariates.

Assuming that the treatment has two levels (denoted by $z = 0, 1$), one approach [12] 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) [13, 14], 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.

Given a binary response variable $Y = 0, 1$, the *virtual twins* (VT) method [2] first uses a random forest [15] model, with $Z, X_1, \ldots, X_M, ZX_1, \ldots, ZX_M, (1-Z)X_1, \ldots, (1-Z)X_M$ as split variables to estimate the treatment effect $\tau = P(Y = 1 \mid Z = 1) - P(Y = 1 \mid Z = 0)$ of each subject. Categorical variables are converted to dummy 0–1 variables for splitting. Then, *recursive partitioning and regression trees* (RPART) [16] is used to construct a classification or regression tree model to predict τ for each subject and to obtain the subgroups. If a classification tree is used, the two classes are defined by the estimated τ being greater or less than a prespecified constant; if a regression tree is used, the subgroups are the terminal nodes with estimated τ greater than a prespecified constant. Although the basic idea is independent of random forest and RPART, their use results in VT inheriting all their weaknesses, such as variable selection bias and (for random forest) lack of a preferred way to deal with missing values.

The *subgroup identification based on differential effect search* (SIDES) method [3] 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 $\{a_j < X_j \leqslant b_j\}$.

Most methods can control the minimum node sample size so that the subgroups have sufficient numbers of observations. The *qualitative interaction tree* (QUINT) method [17] 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. Like the aforementioned methods, QUINT finds the subgroups by searching over all possible splits on all predictor variables and hence is subject to selection bias in its splits. QUINT is currently limited to ordinal X_i and uncensored Y variables. All the methods are limited to two-level treatment variables.

3. Uncensored data

It is well known that evaluating all possible splits on all variables to optimize an objective function leads to a bias toward selecting variables that allow more splits [18–20]. 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 more splits has a greater chance to be selected than one with fewer 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 chi-squared automatic interaction detector (CHAID) [21] classification tree algorithm, which is biased toward selecting variables with few splits.

The generalized unbiased interaction detection and estimation (GUIDE) algorithm [19, 22] 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 split on each of the other variables. It also eliminates selection bias, at least in principle, by using chi-squared tests to select the split variable. QUEST [18], CRUISE [23], CTREE [24], and MOB [25] are other algorithms that employ significance tests for variable selection. In this section, we introduce three ways to 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 [22] 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

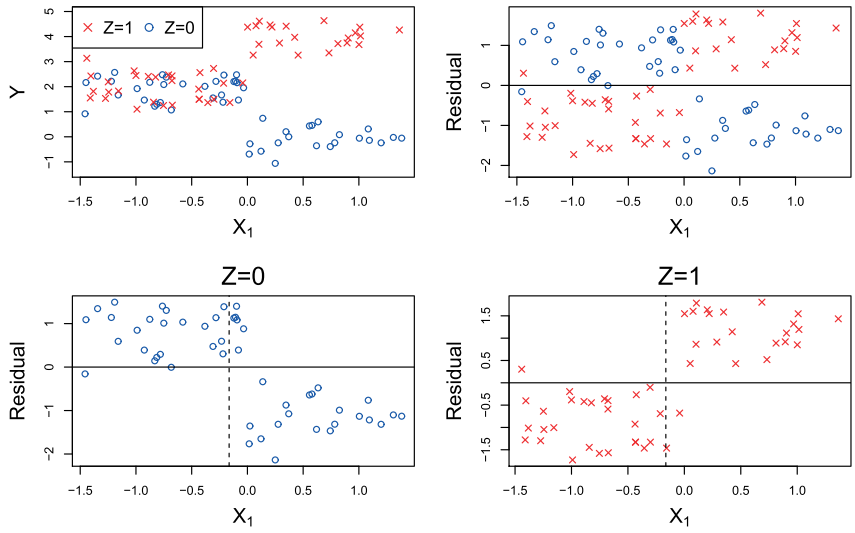
GUIDE linear regression tree [19] offers an alternative approach that permits Y and Z to take more than two values each. At each node, we fit a model linear in Z (with dummy coding) and select a variable to split by examining the residual pattern for each level of Z. Suppose, for example, that data $(X_1, X_2, \ldots, Y, Z)$ are generated from the model

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

where $Z = 0, 1$, and ε is independent normal. The true subgroup being $X_1 > 0$, we should split the data using X_1. Figure 2 shows how this conclusion can be reached by looking at the data. The top row plots Y and the residuals from fitting the model $EY = \beta_0 + \beta_1 Z$. The middle row plots the residuals versus X_1 for each level of Z. The opposite trends in the residual plots indicate that X_1 has an interaction with Z. No other X variable shows such strong patterns. We can quantify 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 the bottom row of the figure and summing the chi-squared statistics over the levels of Z. By applying this procedure to each X_i, we can 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 [26] to convert each contingency table chi-squared statistic to a one-df chi-squared quantile. Specifically, let x and y be chi-squared quantiles with ν df and μ df, respectively. Then [22],

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



$Z=0$	$X_1 \leqslant \bar{x}_1$	$X_1 > \bar{x}_1$
resid > 0	21	6
resid $\leqslant 0$	2	21

$\chi^2 = 21.2,\ p = 4 \times 10^{-6}$

$Z=1$	$X_1 \leqslant \bar{x}_1$	$X_1 > \bar{x}_1$
resid > 0	1	21
resid $\leqslant 0$	26	2

$\chi^2 = 35.2,\ p = 3 \times 10^{-9}$

Figure 2. Plots of Y and residuals versus 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 approximation provides a ranking of the variables without the need for chi-squared p-values that can be extremely small and hard to compute accurately. 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. One distinct advantage of contingency tables is that missing values can be dealt with easily as detailed in the following algorithm.

Algorithm 1
Gs split selection.

(1) Fit the least squares model

$$EY = \beta_0 + \sum_{z=1}^{L-1} \beta_z I(Z = z) \tag{3}$$

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 $z = 1, 2, \ldots, L$.

(a) Form a contingency table from the data in S_z using the signs (positive vs nonpositive) 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' row.

(b) Compute the chi-squared statistic W_z for testing independence, and let ν_z denote its df. Use (2) to convert W_z to the one-df chi-squared quantile

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

(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 one-df chi-squared quantile

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

(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 $A \cap \{X^* \leqslant c\}$ or $\bar{A} \cap \{X^* \leqslant c\}$ that minimizes the sum of the squared residuals of model (3) 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 $(2^{g-1} - 1)$ splits of the form $X^* \in S$ to find the one that minimizes the sum of squared residuals in the two child nodes. If $g \geqslant 10$, limit the search to $(g - 1)$ splits by following a technique in [27, p. 101] for piecewise constant least-squares regression 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.

If there are no missing X^* values in the training data, missing X^* in cases to be predicted are imputed with the node mean.

Unlike the basic GUIDE algorithm [19] which divides the values of each ordered X variable into three or four groups in step 1, we use only two groups here to avoid small cell counts, because each contingency table is constructed from the data for one treatment level only.

The MOB [25] method can be employed similarly, by fitting a linear model as in step 1 of the aforementioned algorithm and then using its own permutation-based procedures to select the split variables and split values. Because Gs and MOB are sensitive to both prognostic and predictive variables, however, they may be ineffective if we wish to find subgroups defined by predictive variables only. To see this, suppose now that the data $(X_1, X_2, \ldots, Y, Z)$ are generated from the true model

$$Y = 2I(Z = 1) + I(X_1 > 0) + \varepsilon \tag{4}$$

with ε independent normal. The simulated data plots in Figure 3 show that Gs and MOB 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

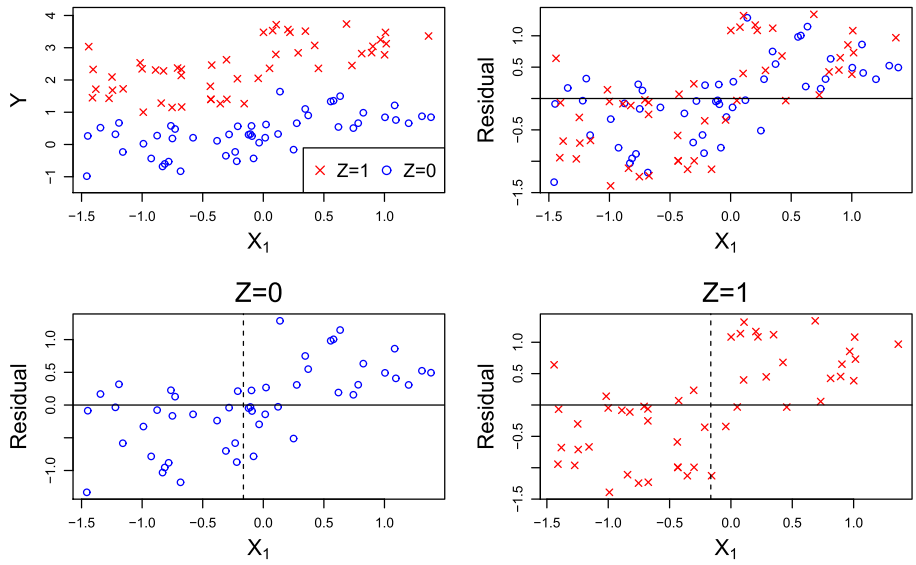


Figure 3. Plots of data and residuals from the model (4), where X_1 is prognostic.

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

$$EY = \beta_0 + \sum_k \beta_k I(Z = k) + \sum_j \gamma_j I(H = j), \tag{5}$$

where $H = X$ if it is categorical and is the indicator function $I(X \leqslant \bar{x})$ 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 and by allowing missing values in X. We call this the Gi method ('i' for interaction). The procedure 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 X values if there are any. Let H denote the factor variable created from the groups.
   (b) Carry out a 'pure error' lack-of-fit test (see, e.g., [28, Sec. 4.3]) of the model (5) 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 1 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.

Applying the method to the data in Figures 2 and 3 yields p-values of 3×10^{-19} and 0.07, respectively.

4. Bias and accuracy

4.1. Selection bias

It is obviously important that a tree model does not 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 property. 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 1–4, and equi-probable categorical with three and seven levels, as shown in Table I.

Based on 2500 simulation iterations with a sample size of 100 in each iteration, Figure 4 shows the frequency that each method selects X_1 to split the root node. 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 toward selecting the variable that has more splits while SIDES and VT are the opposite. In contrast, the selection frequencies of Gc, Gs, and MOB are all within three simulation standard errors. The frequencies of Gi are also within three standard errors, except when X_2 is categorical with seven 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 $P(Z = 0) =$

Table I. Four types of distributions of X_1 and X_2.

Notation	Type	Distribution
Cont	Continuous	Standard normal
Ord4	Ordinal	Discrete uniform with four levels
Cat3	Categorical	Discrete uniform with three levels
Cat7	Categorical	Discrete uniform with seven levels

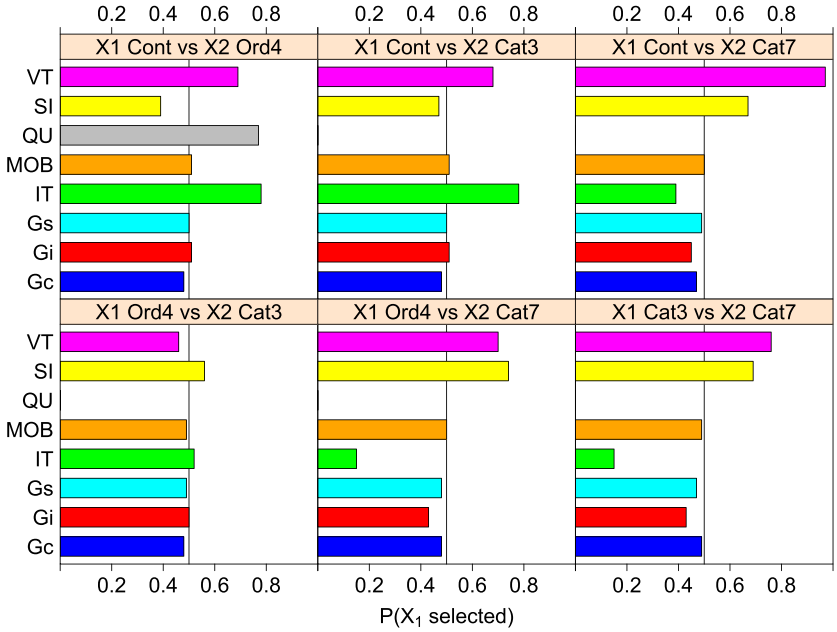


Figure 4. Simulated probabilities that X_1 is selected to split the root node; simulation standard errors less than 0.01. A method is unbiased if it selects X_1 with probability 0.50. SI and QU refer to SIDES and QUINT, respectively. The latter is not applicable to categorical variables.

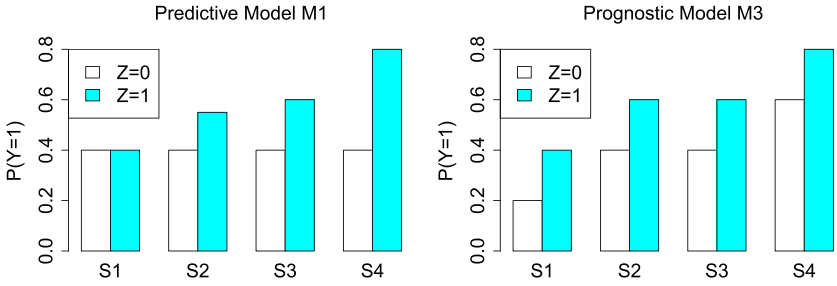


Figure 5. Models M1 and M3. Sets S1 = $\{X_1 = 0, X_2 = 0\}$, S2 = $\{X_1 = 0, X_2 > 0\}$, S3 = $\{X_1 > 0, X_2 = 0\}$, and S4 = $\{X_1 > 0, X_2 > 0\}$.

$P(Z = 1)$ and 100 variables $\mathbf{X} = (X_1, X_2, \ldots, X_{100})$, 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 π_j $(j = 3, 4, \ldots, 100)$ independently simulated from a beta distribution with density $f(x) \propto x(1-x)^2$. The models for Y are as follows:

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

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

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

Figure 5 shows the values of $P(Y = 1 \mid \mathbf{X})$ 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 prognostic and predictive variables.

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

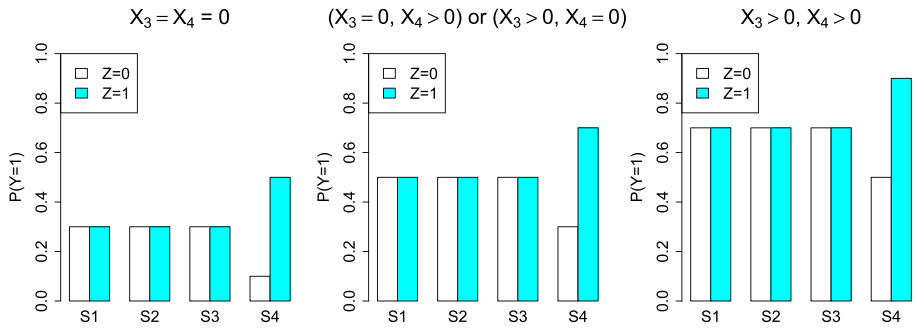


Figure 6. Model M2, with predictive X_1 and X_2 and prognostic X_3 and X_4. Sets S1 $= \{X_1 = 0, X_2 = 0\}$, S2 $= \{X_1 = 0, X_2 > 0\}$, S3 $= \{X_1 > 0, X_2 = 0\}$, and S4 $= \{X_1 > 0, X_2 > 0\}$.

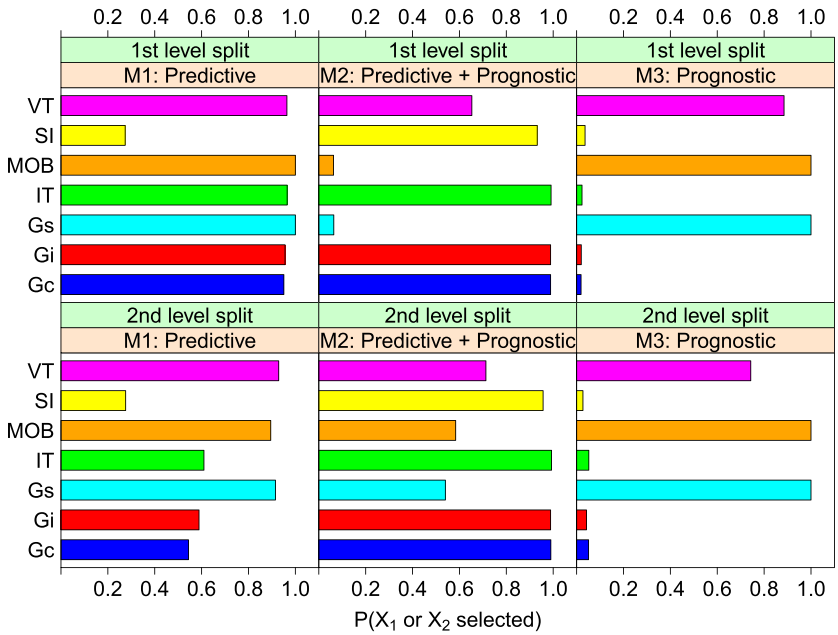


Figure 7. Probabilities that X_1 or X_2 are selected at first and second level splits of trees for models M1, M2, and M3. Long bars are better for M1 and M2, and short bars are better for M3. VT, virtual twins; SI, subgroup identification based on differential effect search; IT, interaction trees.

model, and a tree is constructed using each method. The frequencies that X_1 or X_2 is selected to split the root node (first 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. The AIC pruning penalty is used instead of the default BIC for IT because the AIC yields trivial trees less often.

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 of Gs, MOB, and VT are distinctly higher than those of Gc, Gi, and IT, while that of SIDES remains low. Therefore, Gs, MOB, 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 two predictive and two prognostic variables. Now, Gc, Gi, and IT are excellent with SIDES close behind, and Gs and MOB are very poor. This shows that the latter two do not distinguish between predictive and prognostic variables. This is confirmed 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, MOB, and VT pick the former with high frequencies. Only Gc, Gi, IT, and SIDES perform reliably 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 Section that the effect size is $R(S) = |P(Y = 1 \mid Z = 1, S) - P(Y = 1 \mid Z = 0, S)|$. 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 constant.

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, we take their union. The 'accuracy' of $\hat{S}$ is defined to be $P(\hat{S})/P(S^*)$ if $\hat{S} \subset S^*$ and zero otherwise.

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

Model M1. MOB, VT, Gs, and Gi are best, in that order. IT and SIDES have very low accuracy, because of 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, followed by Gc and SIDES at 0.86 and 0.82, respectively. Gs, MOB, IT, and VT have difficulty distinguishing predictive from prognostic variables; all yield nontrivial trees almost all the time, except for IT which gives a nontrivial tree 56% of the time.

Table II. Accuracy rates of subgroup selection and frequencies of nontrivial trees.

Model	Type	Gi	Gs	Gc	IT	SIDES	VT
M1	Accuracy	0.322	0.430	0.150	0.015	0.024	0.465
M1	P(nontrivial tree)	0.953	0.983	0.990	0.105	0.232	1.000
M2	Accuracy	0.913	0.204	0.855	0.280	0.819	0.430
M2	P(nontrivial tree)	0.979	0.999	1.000	0.562	0.988	1.000
M3	Accuracy	0.939	0.285	0.519	0.886	0.848	0.279
M3	P(nontrivial tree)	0.104	1.000	0.693	0.133	0.410	1.000

IT, interaction trees; SIDES, subgroup identification based on differential effect search; VT, virtual twins.

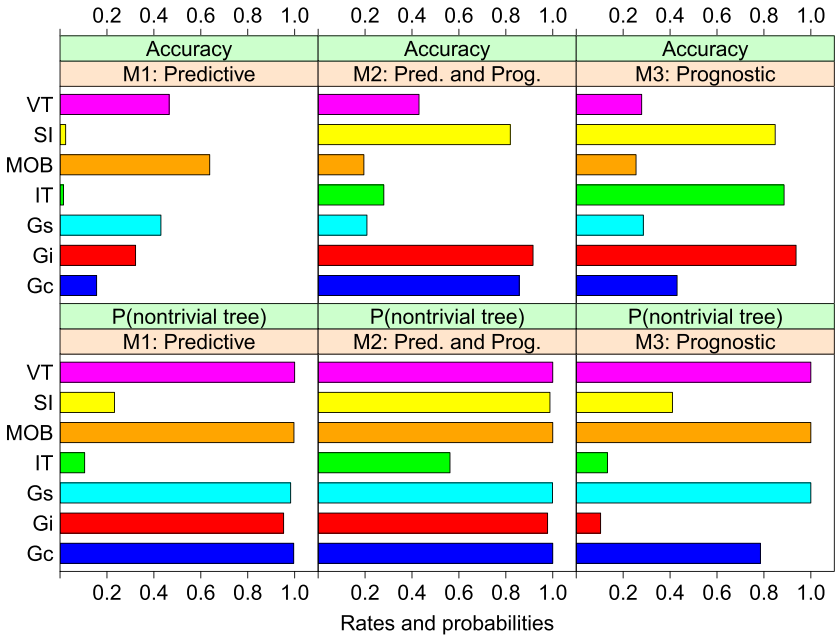


Figure 8. Accuracy rates of subgroup identification and frequencies of nontrivial trees for models M1, M2, and M3. For accuracy, long bars are better. For frequencies of nontrivial trees, long bars are better for M1 and M2 and short bars are better for M3. VT, virtual twins; SI, subgroup identification based on differential effect search; IT, interaction trees.

Model M3. Because S^* is the whole space, the ideal tree is trivial. Gi and IT are best, yielding trivial trees 90% of the time. In terms of accuracy, Gi, IT, and SIDES are best, having values of 0.94, 0.87, and 0.85, respectively. Gs, MOB, and VT are the worst, because they produce nontrivial trees all the time.

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

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), $i = 1, 2, \ldots, n_t$ 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 $\mu(t,z)$. 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 $\mu(t,z)$. If Z takes only two values, 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. Because the nodes in the tree are not fixed in advance but are typically produced by a complex search procedure, however, the validity of these estimates cannot 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 and Z independent Bernoulli with probability 0.50. The binary response Y is 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 terminal nodes t and the proportions of times each naïve confidence interval contains the true estimand. Columns 3–8 of Table III show the estimated bias and coverage probabilities of the intervals over 2000 simulation trials that result in nontrivial trees. The biases are remarkably small (the true means range from 0.30 to 0.90). We attribute this to Gi and Gs not finding splits that directly 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.

To obtain intervals with better coverage, we use a bootstrap method 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 $\mathcal{L}_j^*$ ($j = 1, 2, \ldots, J$) be a bootstrap training sample from $\mathcal{L}$, and let T_j^* be the tree constructed from

Table III. 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 2000 simulations trials of nontrivial trees, with 100 bootstrap iterations per trial. r is the number of replicates of a 3^4 design.

		Bias of naïve estimates of means and difference			Coverage probabilities of 95%					
					naïve intervals			bootstrap intervals		
r	Expt	$\mu(t,0)$	$\mu(t,1)$	$d(t)$	$\mu(t,0)$	$\mu(t,1)$	$d(t)$	$\mu(t,0)$	$\mu(t,1)$	$d(t)$
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

$\mathcal{L}_j^*$ 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

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

The bootstrap estimate of the variance of $\hat{\mu}(t, z)$ is the sample variance $s_\mu^2(t, z)$ of $\{\bar{\mu}_1^*(t, z), \bar{\mu}_2^*(t, z), \ldots, \bar{\mu}_J^*(t, z)\}$ and a 95% bootstrap confidence interval for $\mu(t, z)$ is $\hat{\mu}(t, z) \pm 2s_\mu(t, z)$. If Z takes values zero and one, 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 $\{\bar{d}_1^*(t), \bar{d}_2^*(t), \ldots, \bar{d}_J^*(t)\}$.

The rightmost three columns of Table III give the simulated coverage probabilities of the bootstrap intervals using $J = 100$. There is a clear improvement over the naïve intervals. In particular, the coverage probabilities of 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 (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 [12, 13] is problematic because Gs employs chi-squared tests on residuals and their signs. Although there are many definitions of such residuals [29], 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 requires a common estimated baseline cumulative hazard function. 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 $y_i = \min(u_i, s_i)$, $i = 1, 2, \ldots, n$. 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 log likelihood can be expressed as

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

The first term on the right is the kernel of the log likelihood for n independent Poisson variables δ_i with means μ_i, and the second term is independent of the covariates (see, e.g., [30, 31]). If the $\Lambda_0(y_i)$ values are known, the vector β may be estimated by treating the event indicators δ_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 [32, 33], using δ_i as Poisson responses, the treatment indicators as predictor variables, and $\log \Lambda_0(y_i)$ as offset variable. Gi employs log-linear model goodness-of-fit tests [34, 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 [35, 36] 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., [37, p. 361]). The results reported here are obtained using five iterations.

The results of applying these techniques to the breast cancer data were shown earlier in Figure 1. Gi and Gs each splits the data once, at `progrec` $\leqslant 21$ and `pnodes` $\leqslant 3$, 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 beneath the terminal nodes of the trees. They are constructed as for uncensored response data, with the regression coefficient β 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 jth bootstrap sample and tree, for $j = 1, 2, \ldots, J$. Let $\hat{\beta}(t)$ and $\hat{\beta}_j^*(t^*)$ denote the estimates of β 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 $\bar{\beta}_j^*(t) = \sum_{t^*} n(t \cap t^*)\ \hat{\beta}_j^*(t^*) \Big/ \sum_{t^*} n(t \cap t^*)$. The bootstrap estimate of the variance of $\hat{\beta}(t)$ is the sample variance $s_\beta^2(t)$ of $\{\bar{\beta}_1^*(t), \bar{\beta}_2^*(t), \ldots, \bar{\beta}_J^*(t)\}$ 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, but few 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 [38], we score the importance of a variable X in terms of the one-df chi-squared statistics computed during variable selection. Specifically, let $q_t(X)$ be the value of $q(X)$ (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 [39]. This procedure is similar to that in [38] except for two changes. First, the latter employs the weight $\sqrt{n_t}$ 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 [38], 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. For the breast cancer data, Gi identifies only `progrec` as important, whereas Gs identifies `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 [23, 40]. 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 [41].

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 63% censored. The explanatory variables consist of 17 continuous-valued baseline measures (`a1`, `a2`, and `b01`–`b15`) and 288 categorical variables, of which six are baseline measures (`c0`–`c5`) and the rest are genetic variables (`g001`–`g282`), each having two or three levels. More than 95% (1435/1504) of the subjects have values missing in one or more explanatory variables; only seven variables (`a1`, `a2`, `b3`, `c0`, `c4`, `b15`, and `g272`) are completely observed.

Although the overall treatment effect is statistically significant (p-value 0.008), its magnitude is small. The question is whether there is a subgroup for which the treatment effect is larger. 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 shown by the Kaplan–Meier

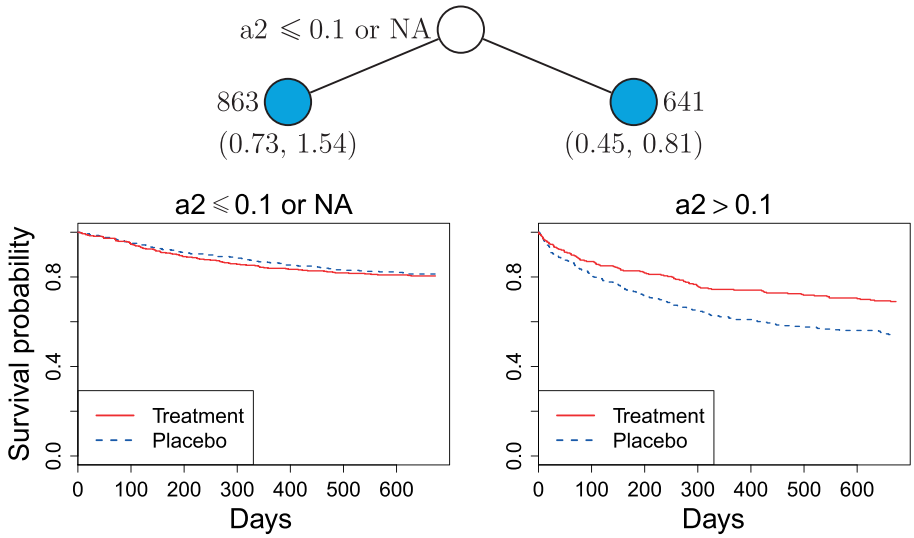


Figure 9. Gs model for gene data. At each node, a case goes to the left child node if and only if the stated condition is satisfied. Sample sizes are beside terminal nodes and 95% bootstrap intervals for relative risk of recurrence for treatment versus placebo are below the nodes.

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-valued response and treatment variables. Further, some modification of the classification tree algorithm is needed to disallow splits that yield nodes with pure classes.

Gs is a direct 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 multivalued treatment variables, and to all types of predictors variables, with or without missing values. 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 negatively affected 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 allows missing predictor values and saves much computation time.

We extend Gi and Gs to censored time-to-event data by fitting a tree-structured proportional hazards model to the data using Poisson regression. Poisson residuals are easier to employ for our purposes than those from proportional hazards models. Further, this approach gives a common baseline cumulative hazard function, thereby allowing comparisons of treatment effects between nodes. The price is increased computing time because of 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

Table IV. Average times (s), over 500 simulation trials of model M1, to construct one tree on a 2.66 GHz Intel processor.

MOB	Gs	Gi	Gc	IT	VT	SI	QU
1.4	4.3	7.0	17.5	127.8	341.1	1601.5	NA

QU does not allow categorical variables.
IT, interaction trees; VT, virtual twins; SI, subgroup identification based on differential effect search; QU, qualitative interaction tree.

times to construct one tree for model M1 in Table IV (recall that the predictor variables in M1 take three values each; the speeds of Gc, Gi, and Gs relative to the other methods would be greater if the variables take more values).

Subgroup identification may be prone to error, especially if the number of predictor variables is large, because the chance of finding the correct variables can be small, as the results for model M1 in Figure 8 show. 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 methods discussed here cannot be expected to provide definitive subgroup identification, because the error rates can be quite high in some situations (e.g., model M1). Of course, it would increase confidence if there is a procedure that can indicate the magnitude of the error rate. Until then, the main utility of the methods is in identifying potential subgroups for scrutiny by domain knowledge experts and validation by independent trials.

Gi and Gs are implemented in the GUIDE computer program which can be obtained from www.stat.wisc.edu/~loh/guide.html.

Acknowledgements

We are grateful to Xiaogang Su, Jared Foster, Jue Hou, Elise Dusseldorp, and Achim Zeileis for sharing with us their R programs for IT, VT, SIDES, QUINT, and MOB, respectively, and for their patience in answering our questions. We also thank Lei Shen and an anonymous referee for helpful comments on the manuscript. This work was partially supported by the US 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. Ciampi A, Negassa A, Lou Z. Tree-structured prediction for censored survival data and the Cox model. *Journal of Clinical Epidemiology* 1995; **48**:675–689.
2. Foster JC, Taylor JMG, Ruberg SJ. Subgroup identification from randomized clinical trial data. *Statistics in Medicine* 2011; **30**:2867–2880.
3. Lipkovich I, Dmitrienko A, Denne J, Enas G. Subgroup identification based on differential effect search — a recursive partitioning method for establishing response to treatment in patient subpopulations. *Statistics in Medicine* 2011; **30**: 2601–2621.
4. Schmoor C, Olschewski M, Schumacher M. Randomized and non-randomized patients in clinical trials: experiences with comprehensive cohort studies. *Statistics in Medicine* 1996; **15**:263–271.
5. Peters A, Hothorn T. *Improved predictors*, 2012. R package version 0.8-13.
6. Schumacher M, Baster G, Bojar H, Hübner K, Olschewski M, Sauerbrei W, Schmoor C, Beyerle C, Newmann RLA, Rauschecker HF. Randomized 2 × 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. *Journal of Clinical Oncology* 1994; **12**:2086–2093.

7. Sauerbrei W, Royston P. Building multivariable prognostic and diagnostic models: transformation of the predictors by using fractional polynomials. *Journal of the Royal Statistical Society, Series A* 1999; **162**:71–94.
8. Everitt BS, Hothorn T. *A Handbook of Statistical Analyses Using R*. Chapman and Hall/CRC: New York, 2006.
9. Mehta S, Shelling A, Muthukaruppan A, Lasham A, Blenkiron C, Laking G, Print C. Predictive and prognostic molecular markers for cancer medicine. *Therapeutic Advances in Medical Oncology* 2010; **2**(2):125–148.
10. Italiano A. Prognostic or predictive? It's time to get back to definitions! *Journal of Clinical Oncology* 2011; **29**:4718.
11. Su XG, Kang J, Fan J, Levine R, Yan X. Facilitating score and causal inference trees for large observational data. *Journal of Machine Learning Research* 2012; **13**:2955–2994.
12. Negassa A, Ciampi A, Abrahamowicz M, Shapiro S, Boivin JR. Tree-structured subgroup analysis for censored survival data: validation of computationally inexpensive model selection criteria. *Statistics and Computing* 2005; **15**:231–239.
13. Su X, Zhou T, Yan X, Fan J, Yang S. Interaction trees with censored survival data. *International Journal of Biostatistics* 2008; **4**. Article 2. DOI: 10.2202/1557-4679.1071
14. Su X, Tsai CL, Wang H, Nickerson DM, Bogong L. Subgroup analysis via recursive partitioning. *Journal of Machine Learning Research* 2009; **10**:141–158.
15. Breiman L. Random forests. *Machine Learning* 2001; **45**:5–32.
16. Therneau TM, Atkinson B. *Rpart: recursive partitioning*, 2012. R package version 3.1-51.
17. Dusseldorp E, VanMechelen I. Qualitative interaction trees: a tool to identify qualitative treatment-subgroup interactions. *Statistics in Medicine* 2014; **33**:219–237.
18. Loh WY, Shih YS. Split selection methods for classification trees. *Statistica Sinica* 1997; **7**:815–840.
19. Loh WY. Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica* 2002; **12**:361–386.
20. Loh WY. Fifty years of classification and regression trees (with discussion). *International Statistical Review* 2014; **34**: 329–370.
21. Kass GV. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics* 1980; **29**: 119–127.
22. Loh WY. Improving the precision of classification trees. *Annals of Applied Statistics* 2009; **3**:1710–1737.
23. Kim H, Loh WY. Classification trees with unbiased multiway splits. *Journal of the American Statistical Association* 2001; **96**:589–604.
24. Hothorn T, Hornik K, Strobl C, Zeileis A. *Party: a laboratory for recursive partytioning*, 2012. R package version 1.0-1.
25. Zeileis A, Hothorn T, Hornik K. Model-based recursive partitioning. *Journal of Computational and Graphical Statistics* 2008; **17**:492–514.
26. Wilson EB, Hilferty MM. The distribution of chi-square. *Proceedings of the National Academy of Sciences of the United States of America* 1931; **17**:684–688.
27. Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and Regression Trees*. Wadsworth: Belmont, California, 1984.
28. Weisberg S. *Applied Linear Regression* 2nd ed. Wiley: New York, 1985.
29. Ahn H, Loh WY. Tree-structured proportional hazards regression modeling. *Biometrics* 1994; **50**:471–485.
30. Aitkin M, Clayton D. The fitting of exponential, Weibull and extreme value distributions to complex censored survival data using GLIM. *Applied Statistics* 1980; **29**:156–163.
31. Laird N, Olivier D. Covariance analysis of censored survival data using log-linear analysis techniques. *Journal of the American Statistical Association* 1981; **76**:231–240.
32. Chaudhuri P, Lo WD, Loh WY, Yang C-C. Generalized regression trees. *Statistica Sinica* 1995; **5**:641–666.
33. Loh WY. Regression tree models for designed experiments. In *Second E. L. Lehmann Symposium*, Rojo J (ed.), Institute of Mathematical Statistics Lecture Notes-Monograph Series, vol. 49, 2006; 210–228.
34. Agresti A. *An Introduction to Categorical Data Analysis* 2nd ed. Wiley, 2007.
35. Aalen OO. Nonparametric inference for a family of counting processes. *Annals of Statistics* 1978; **6**:701–726.
36. Breslow N. Contribution to the discussion of regression models and life tables by D. R. Cox. *Journal of the Royal Statistical Society, Ser. B* 1972; **34**:216–217.
37. Lawless JF. *Statistical Models and Methods for Lifetime Data*. Wiley: New York, 1982.
38. Loh WY. Variable selection for classification and regression in large p, small n problems. In *Probability Approximations and Beyond*, Barbour A, Chan HP, Siegmund D (eds), Lecture Notes in Statistics—Proceedings, vol. 205. Springer: New York, 2012; 133–157.
39. Satterthwaite FE. An approximate distribution of estimates of variance components. *Biometrics Bulletin* 1946; **2**:110–114.
40. Strobl C, Boulesteix AL, Zeileis A, Hothorn T. Bias in random forest variable importance measures: illustrations, sources and a solution. *BMC Bioinformatics* 2007; **8**:25. DOI: 10.1186/1471-2105-8-25.
41. Ding Y, Simonoff JS. An investigation of missing data methods for classification trees applied to binary response data. *Journal of Machine Learning Research* 2010; **11**:131–170.