Research Article

Statistics in Medicine

Received 2 November 2015, Accepted 31 May 2016 Published online 27 June 2016 in Wiley Online Library

(wileyonlinelibrary.com) DOI: 10.1002/sim.7020

Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables

Wei-Yin Loh,[a*†] Haoda Fu,[b] Michael Man,[b] Victoria Champion[c] and Menggang Yu[d]

We describe and evaluate a regression tree algorithm for finding subgroups with differential treatments effects in randomized trials with multivariate outcomes. The data may contain missing values in the outcomes and covariates, and the treatment variable is not limited to two levels. Simulation results show that the regression tree models have unbiased variable selection and the estimates of subgroup treatment effects are approximately unbiased. A bootstrap calibration technique is proposed for constructing confidence intervals for the treatment effects. The method is illustrated with data from a longitudinal study comparing two diabetes drugs and a mammography screening trial comparing two treatments and a control. Copyright © 2016 John Wiley & Sons, Ltd.

Keywords: bootstrap; precision medicine; randomized trial; regression tree; unbiased

1. Introduction

Interest in precision medicine (also known as *personalized medicine* and *tailored therapeutics*), where therapies are customized for individual patients based on their genetic and other characteristics, is drawing more attention to regression tree methods designed to identify subgroups with differential treatment effects from randomized trials. The interaction trees [1–3] method selects splits that minimize p-values of interaction terms in models fitted to the nodes of the trees. The virtual twins [4] method estimates the treatment effect of each subject using a random forest [5] model and then fits a CART [6] tree to the estimated effects to obtain the subgroups. SIDES [7] finds multiple subgroups that optimize certain measures (such as p-values or treatment effect sizes). QUINT [8] searches for subgroups that balance effect size and subgroup size. Although obvious and seemingly natural, algorithms that search for splits that optimize one or more criteria have two undesirable consequences: *biased variable selection* (all things being equal, some variables are more likely than others to be selected to define the subgroups) and *biased estimates of subgroup effects* (estimated differences in treatment effects between subgroups are overly large). Loh *et al*. [9] extended the GUIDE [10–12] approach to find subgroups without these biases. Except for Su *et al*. [3], the methods are applicable to a single outcome variable only. The purpose of this article is to further extend the GUIDE subgroup identification approach to multivariate outcome variables.

To illustrate, consider a multi-center, randomized double-blind trial on the long-term efficacy and safety of *Pioglitazone* versus *Gliclazide* in patients with Type 2 diabetes mellitus that is inadequately controlled by diet alone [13]. Gliclazide increases the amount of insulin produced by the pancreas while Pioglitazone is an 'insulin sensitizer'—it improves the ability of the body to use insulin. The trial consisted of 1249 subjects between 35 and 75 years old with HbA1c between 7.5% and 11.0% and for whom diet was prescribed for at least 3 months. Each subject was randomized to a 52-week treatment period consisting of a 16-week forced-titration period to a maximum dose and a 36-week maintenance period

[a]*Department of Statistics, University of Wisconsin, Madison, WI 53706, U.S.A.*
[b]*Eli Lilly Company, Indianapolis, IN 46285, U.S.A.*
[c]*School of Nursing, Indiana University, Indianapolis, IN 46202, U.S.A.*
[d]*Department of Biostatistics & Medical Informatics University of Wisconsin, Madison, WI 53706, 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*

Table I. Baseline predictor variables for diabetes data.

Variable	#Missing		Variable	#Missing	
HDL	7	32	Age	0	0
LDL	77	158	Weight	1	1
Total cholesterol	6	31	BMI	0	0
Triglycerides	6	31	Waist	4	4
Creatinine	0	1	HbA1Cbase	0	0
FastInsulin (fasting insulin)	46	142	HOMA-S	62	172
ALT (alanine aminotransferase)	0	2	HOMA-IR	62	172
AST (aspartate aminotransferase)	0	2	HOMA-B	62	172
GGT (γ-glutamyl transpeptidase)	0	1	Diastolic blood pressure	0	0
C-peptide	593	985	Systolic blood pressure	0	0
Diabetes duration	0	0	Pulse	0	1
FastBG (fasting blood glucose)	0	0			

The missing value columns pertain to the subset of 747 subjects with complete outcome variables and to the full set of 1249 subjects.
HOMA, Homeostasis Model Assessment; B, beta cell function; IR, insulin resistance; S, insulin sensitivity.

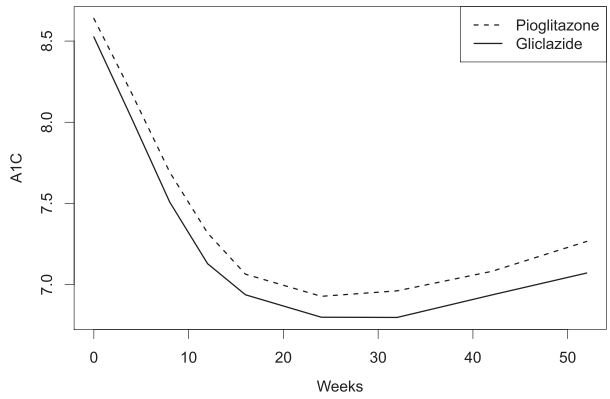

Figure 1. HbA1c means for Pioglitazone and Gliclazide.

at the maximum tolerated dose of the drug. The treatments were 80mg Gliclazide (625 subjects), 30mg Pioglitazone (114 subjects), and 45mg Pioglitazone (510 subjects). Twenty-three baseline variables were measured for each subject. There are nine outcome variables, namely, HbA1c at 0, 4, 8, 12, 16, 24, 32, 42, and 52 weeks. The primary efficacy endpoint is change from baseline HbA1c.

Combining the subjects given 30 and 45 mg Pioglitazone into one 'Pioglitazone' group gives 747 subjects (383 and 364 in the Pioglitazone and Gliclazide groups, respectively) with complete HbA1c values at all time points. Table I gives the names, definitions, and numbers of missing values of the predictor variables, and Figure 1 plots the group mean HbA1c values over time. Gliclazide appears to be better, on average, than Pioglitazone in lowering HbA1c throughout. But is there a subgroup for which Pioglitazone might be better for at least some time points? Figure 2 shows one possible subgroup, defined by HOMA-B $>$ 23.90 and FastBG $>$ 10.85, where Pioglitazone appears to control HbA1C better than Gliclazide after 25 weeks.

2. Method

GUIDE is a general classification and regression tree algorithm, and Gi is an option for subgroup identification. We describe in this section how the Gi option is extended to obtain the tree in Figure 2. First, we review the method for the case of one outcome variable, mentioning improvements since its introduction in Loh *et al*. [9].

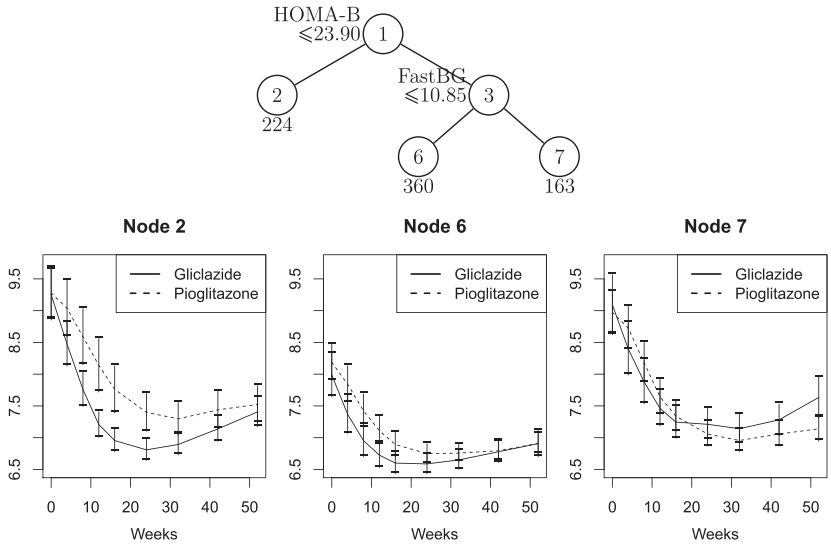

Figure 2. GUIDE tree for diabetes data with plots of mean HbA1C, using LDA. Error bars are 95% bootstrap confidence intervals. Sample sizes printed beneath nodes.

2.1. One outcome variable

A unique feature of Gi is how it selects a split of the data in each node of a tree. Let Y denote the (single) outcome variable and Z a treatment variable taking G nominal values $z = 1, 2, \ldots, G$. Let X_i be a predictor variable. At each node t of the tree, a lack-of-fit F test is used to select an X_i to split the data in t. If X_i is an ordinal variable, the test temporarily converts it into a two-group categorical variable H_i by splitting its values at the mean. If X_i is categorical, then $H_i = X_i$ with each category forming a group. If there are missing values in X_i, a 'missing' group is added. This allows observations with missing values to be included for variable selection at every node.

We fit the additive model $EY = \beta_0 + \sum_{z=1}^{G-1} \beta_z I(Z = z) + \sum_h \gamma_h I(H_i = h)$ to the data in t and obtain its F-statistic F_i and p-value p_i for the 'pure error' lack-of-fit test [14, Sec. 4.3]. Our goal is to select the most significant X_i to split the data in the node. The value of p_i can be tiny and hard to compute if X_i has a large interaction with Z. To avoid its computation in such situations, we transform the F_i statistics to 1-df chi-squared quantiles and select the X_i with the largest chi-squared instead. Let ν_i and μ_i be the numerator and denominator dfs of F_i, and let φ_i and τ_i^2 denote the mean and variance, respectively, of the central F distribution with these dfs. Transformation of F_i to chi-squared is carried out in two parts.

(1) If F_i is not extremely large (specifically, $\mu_i < 10$ and $F_i < 3000\tau_i + \varphi_i$ or $\mu_i \geqslant 10$ and $F_i < 150\tau_i + \varphi_i$), compute p_i directly from the F distribution and then compute the $(1 - p_i)$-quantile $\chi_1^2(i)$ of the chi-squared distribution with 1 df.
(2) Otherwise, use a two-step approximation:
   (a). Compute $a = \nu_i F_i/3$ and $b = (2\mu_i + a + \nu_i - 2)/\{2(\mu_i + 2a)\}$. Then $\chi^2_{\nu_i} = b\nu_i F_i$ is approximately the $(1 - p_i)$-quantile of a chi-squared distribution with ν_i df [15].
   (b). Compute

$$w_1 = \left\{ \sqrt{2\chi^2_{\nu_i}} - \sqrt{2\nu_i - 1} + 1 \right\}^2 / 2$$

$$w_2 = \max\left(0, \left[\frac{7}{9} + \sqrt{\nu_i}\left\{\left(\frac{\chi^2_{\nu_i}}{\nu_i}\right)^{1/3} - 1 + \frac{2}{9\nu_i}\right\}\right]^3\right)$$

$$\chi_1^2(i) = \begin{cases} w_2 & \text{if } \chi^2_{\nu_i} < \nu_i + 10\sqrt{2\nu_i} \\ (w_1 + w_2)/2 & \text{if } \chi^2_{\nu_i} \geqslant \nu_i + 10\sqrt{2\nu_i} \text{ and } w_2 < \chi^2_{\nu_i} \\ w_1 & \text{otherwise} \end{cases}$$

Then $\chi^2_1(i)$ is approximately the $(1-p_i)$-quantile of a chi-squared distribution with 1 df. The result is obtained by combining two approximations in Wilson and Hilferty [16] (see [17, p. 427]). If $\nu_i = 1$, then $w_1 = w_2 = \chi^2_{\nu_i}$ and $\chi^2_1(i) = \chi^2_{\nu_i}$, and this step is not needed.

Part 2(b) improves upon a earlier approximation used in Loh [11] and Loh *et al*. [9].

Let X^* be the variable with the largest value of $\chi^2_1(i)$. The data in t are partitioned into left and right child nodes by a split on X^* of the form '$X^* \leqslant c$' if X^* is ordinal or '$X^* \in C$' where C is a subset of the values of X^* if it is categorical. The best split is found by minimizing $S_L + S_R$, where S_L and S_R are the sums of the squared residuals of the treatment-only model $EY = \beta_0 + \sum_z \beta_z I(Z = z)$ fitted independently to the left and right child nodes. Only splits that yield child nodes with all treatment levels present are considered. The whole procedure is applied recursively at each node until the number of observations is below a threshold (e.g., 5% of the sample size). Then, the tree is pruned using the CART method, and 10-fold cross-validation (CV) is used to find the smallest CV mean squared error MSE_0 among the subtrees. The smallest subtree with CV MSE within half a standard error of MSE_0 is selected.

What happens if there are missing X^* values in the training sample or in a future sample to be predicted? Following [9, 11], we send all missing X^* values to the same child node. Let `NA` denote the missing value code, and let t_L denote the left child node of a split of t on X^*. If X^* is an ordinal variable, the minimization of $S_L + S_R$ is over c for t_L of the form (i) '$X^* = \texttt{NA}$', (ii) '$X^* \leqslant c$ and $X^* = \texttt{NA}$', and (iii) '$X^* \leqslant c$ and $X^* \neq \texttt{NA}$'. If X^* is categorical, its missing values are treated as another category. Other algorithms use different methods to deal with missing values. CART uses surrogate splits, but they do not give better predictions. Another approach is imputation of the missing values using, for example, regression of X^* on the other X variables, but this may not work well if the other X variables have missing values too [18]. There is one esthetic advantage to sending all missing values to one child node: the split can be displayed compactly in a tree diagram. To indicate that missing values, go to the left child node in a split on an ordinal X; we add an asterisk subscript to the inequality sign, for example, '$X \leqslant_* c$'. If missing values, go to the right child node; we denote the split simply as '$X \leqslant c$'. See Figure 3 later for an example.

2.2. *Multiple outcome variables*

The method for one outcome can be extended to more than one outcome by applying it to one Y_j at a time. For each X_i, we now have a Wilson–Hilferty 1-df chi-squared value, $\chi^2_1(i,j)$ say, for each Y_j. Let $q_i = \sum_j \chi^2_1(i,j)$ be the sum of the chi-squared values over the outcomes. Let X^* be the X_i for which q_i is maximum. Then, for each binary split of the data on X^* in t, we fit the model

$$EY_j = \beta_{0j} + \sum_{z=1}^{G-1} \beta_{zj} I(Z = z), \quad j = 1, 2, \ldots \tag{1}$$

to the data in the two subnodes and choose the split that minimizes the total sum of the squared residuals, where the total is over the outcomes and the two subnodes. (We considered using $\max_j \chi^2_1(i,j)$ in place of q_i but the results are not as good. Besides, a sum of chi-squared has advantages in importance scoring of variables, a topic not discussed here.)

The power and versatility of this approach can be improved with two additions:

Local linear transformations. One weakness of the technique is that it ignores correlations among the Y_j. A standard solution is to transform $(Y_1, Y_2, \ldots)$ to another coordinate system using, for example, principal component analysis (PCA) or linear discriminant analysis (LDA) with Z as the class variable. But this may be ineffective if the correlation structure is not constant over the predictor variable space. A better solution is to perform PCA or LDA *independently at each node t*. For PCA, this is achieved by replacing the Y_j values with their principal components at the node in the computation of the lack-of-fit tests. The split on the selected variable X^* is found as before, that is, the sums of squared residuals are based on the untransformed Y_j values. The procedure for LDA is the same, except that the Y_j values are replaced by the canonical variates (discriminant coordinates) in the computation of the lack-of-fit tests.

Weights. The Y_j may be measured on different scales; they may not be equally important, or they may measure different outcomes (e.g., Y_1 is a measure of efficacy and Y_2 a measure of safety). If they are measured on different scales, they may be normalized to have equal sample variance prior to analysis. If they are not equally important or they measure different outcomes, a weighted total sum of squared residuals may be used to search for the best split on X^*, with weights chosen by the user.

Algorithm 1 presents the basic procedure in pseudocode.

Data: X_i is the ith predictor variable in node t. Y_j is the jth outcome variable, jth principal component in t, or jth linear discriminant variate in t. Z is the treatment variable taking values $k = 1, 2, \ldots, G$.
Result: Split s^* of node t
begin
  foreach i *with non-constant* X_i **do**
    if X_i *is ordinal (continuous or discrete)* **then**
      divide its values into two groups at the node sample mean of X_i;
    else
      define the groups by the categorical values of X_i;
    end
    add a group for missing X_i values if there are any;
    $H_i \leftarrow$ factor variable created from the groups;
    foreach Y_j **do**
      fit an additive model to Y_j using only H_i and Z;
      perform lack-of-fit test and find 1-df chi-squared statistic $\chi^2_1(i, j)$;
    end
    $q_i \leftarrow \sum_j \chi^2_1(i, j)$;
  end
  $i^* \leftarrow \arg\min q_i$;
  foreach *split* s *of* t *on* X_{i^*} **do**
    foreach Y_j **do**
      fit model $EY_j = \eta_j + \sum_{k=1}^{G-1} \beta_{j,k} I(Z = k)$ to each child node;
      let u_j be the total sum of squared residuals in the two child nodes of s;
    end
    $v(s) \leftarrow \sum_j u_j$;
  end
  return $s^* \leftarrow \arg\min v(s)$;
end

Algorithm 1: GUIDE split selection method for multiple outcomes

The result in Figure 2 is obtained with LDA transformation of the observed HbA1c values at each node. The corresponding results without any transformation and with PCA transformation are shown in Figures 3 and 4, respectively. The model with PCA is a subtree of the one with LDA. All three tree models split on HOMA-B at some point. Figure 2 is easiest to interpret based on our understanding of how the drugs work. HOMA-B is a measure of beta cell function, which is the ability to produce insulin. Low values of HOMA-B indicate worse beta cell function. Node 2 in Figure 2 contains subjects with poor beta cell function (less insulin production). Gliclazide works well for these patients because they are insulin deficient. Pioglitazone does not work as well because these patients do not have much insulin in their bodies; making them more sensitive to insulin may not be the best solution. Stimulation of the deteriorating beta cells to produce additional insulin may instead accelerate the decline of beta cell function. At intermediate node 3, where patients have relatively good beta cell function and often good amounts of insulin, Pioglitazone is expected to work better. The split there on FastBG is meaningful. Patients with high FastBG (node 7) often have more insulin in their body conditional on the same beta cell function. Pioglitazone seems to work better for them after 20 weeks. One may wonder why node 3 is split on FastBG instead of FastInsulin. The answer may be because high FastBG is indicative of greater potential for decreasing blood glucose. Therefore, FastBG is an excellent biomarker for high insulin as well as potential for improvement. Figure 5 shows a plot of FastBG versus HOMA-B for the whole data set.

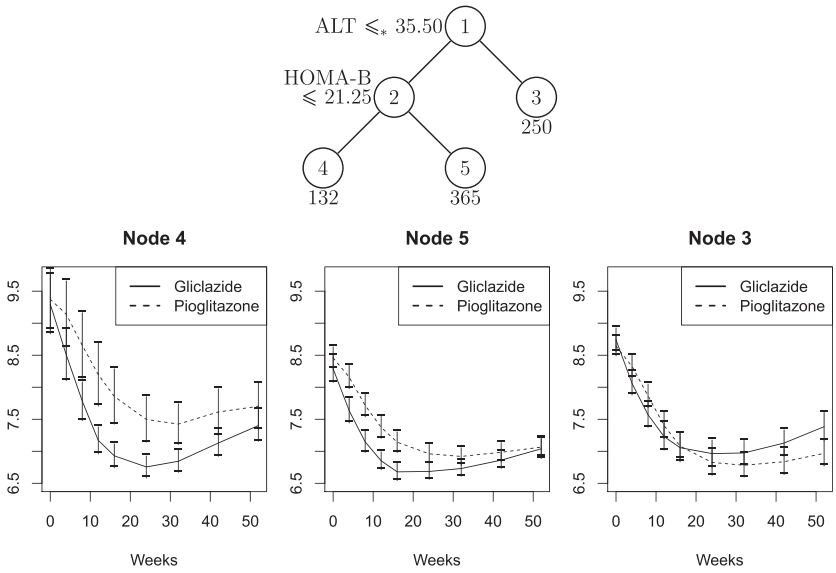

Figure 3. GUIDE tree for diabetes data with plots of mean HbA1C, using neither PCA nor LDA. Error bars are 95% bootstrap confidence intervals. Sample sizes printed beneath nodes. The symbol '⩽*' stands for '⩽ or missing'.

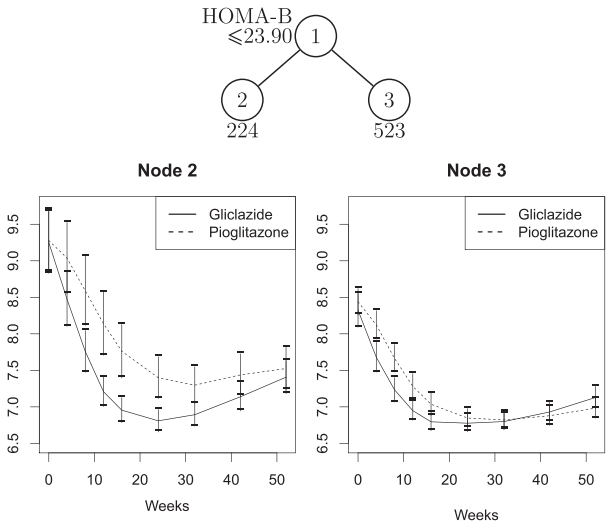

Figure 4. GUIDE tree for diabetes data with plots of mean HbA1C, using PCA. Error bars are 95% bootstrap confidence intervals. Sample sizes printed beneath nodes.

It is less easy to explain why ALT is chosen to split the root node in Figure 3. The split is not counter intuitive, however, because ALT is a biomarker for liver function, with large values indicative of liver damage. Gastaldelli *et al*. [19] found that Pioglitazone works by reducing liver glucose synthesis, and Harris [20] found that the latter is associated with ALT. Using ALT as a predictor, however, is complicated by the fact that men and women have different normal ranges (male <43, female <34) and by gender not being among the variables reported in the data.

3. Simulation results on bias

A tree model needs three essential properties for interpretability: (i) unbiased variable selection; (ii) unbiased estimates of treatment means in the nodes; and (iii) confidence intervals for the treatment means. This section uses simulations to show that the proposed method has the first two properties. The third property is addressed in Section 4.

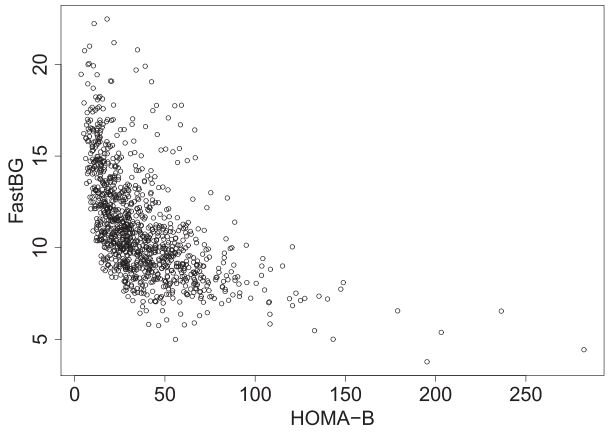

Figure 5. Plot of FastBG versus HOMA-B.

Table II. Estimated probabilities of variable selection from 10,000 simulation iterations.

Model	X_1 $M(2)$	X_2 $M(10)$	X_3 $U(0,1)$	X_4 $E(1)$	X_5 $N(0,1)$
(2)	0.2032	0.2113	0.1924	0.1979	0.1952
(3)	0.1988	0.2081	0.1973	0.1962	0.1996

Standard errors are approximately 0.0040.

3.1. Bias in variable selection

A basic requirement for model interpretation is that the algorithm selects variables to split the nodes without bias. That is, if all the predictor variables are independent of the outcomes, each should be selected with equal probability. CART [6] and algorithms that adopt its paradigm are known to be biased in selecting variables that allow more splits [10, 11, 21, 22]. There are two consequences to biased selection. One is increased likelihood that the subgroups are defined by less relevant variables, which undermines confidence in the conclusions. Another is that if splits on less relevant variables occur sufficiently often in a tree, pruning may remove many of the splits, which reduces the power of the procedure.

To see whether our method is unbiased, we carried out two simulation experiments where the predictor variables are independent of the longitudinal outcomes. Let $M(m)$ denote the multinomial distribution with m equi-probable cells, $U(c,d)$ the uniform distribution on the interval (c,d), $E(\theta)$ the exponential distribution with mean θ, and $N(\mu,\sigma^2)$ the normal distribution with mean μ and variance σ^2. Let X_1, X_2, …, X_5, and Z be mutually independent predictor and treatment variables with distributions $X_1 \sim M(2)$, $X_2 \sim M(10)$, $X_3 \sim U(0,1)$, $X_4 \sim E(1)$, $X_5 \sim N(0,1)$, and $Z \sim M(2)$. Let Y_{ij} denote the outcome for subject i at time j ($j = 1, 2, \ldots, 10$). The first experiment employs a linear mean function and the second a quadratic one that mimics the mean function of the diabetes data:

$$Y_{ij} = b_{1i} + b_{2i}jZ + \epsilon_{ij} \tag{2}$$

$$Y_{ij} = b_{1i} + 0.1(b_{2i} + j + Z - 5)^2 + \epsilon_{ij}. \tag{3}$$

Here $b_{1i} \sim U(3,4)$, $b_{2i} \sim U(0,1)$, and $\epsilon_{ij} \sim N(0,1)$ are mutually independent, representing random subject effects and measurement error, respectively.

Two hundred sample vectors $(X_1, X_2, \ldots, X_5, Y, Z)$ were repeatedly simulated, and the frequency that each X variable was selected to split the root node of the tree was recorded. The results in Table II show the average frequencies over 10,000 simulation trials; all are within three simulation standard errors of 0.20, the target value if variable selection is unbiased.

3.2. Bias in treatment effects

It is equally important for the estimated treatment mean $\mu_t(z)$ for $Z = z$ in the terminal nodes t be unbiased. Many algorithms, such as SIDES [7] and QUINT [8], search for split points that optimize treatment

effects between nodes. As a result, they tend to yield overly optimistic estimates that require subsequent bias adjustment. The difficulty with evaluating the bias of the treatment means in a node t of a tree is that t is not fixed but is a function of the training sample. Loh *et al.* [9] instead estimate the average bias of the means, where the average is over all terminal nodes t. They show by simulation that the bias is remarkably small. Their results are, however, limited to a single outcome and categorical predictor variables with three categories each.

Given a node t in a tree T and $Z = z$, let $\mu_j(z,t) = EY_j$ denote the mean of Y_j for treatment z in t, and let $\hat{\mu}_j(z,t)$ be its estimate. Let $\tilde{T}$ denote the set of terminal nodes of T and $|\tilde{T}|$ be its number of terminal nodes. Define the average error over $t \in \tilde{T}$ for outcome Y_j and treatment level z as $D_j(T,z) = |\tilde{T}|^{-1} \sum_{t\in\tilde{T}} \{\hat{\mu}_j(z,t) - \mu_j(z,t)\}$. Similarly, define the average relative error $R_j(T,z) = |\tilde{T}|^{-1} \sum_{t\in\tilde{T}} \{\hat{\mu}_j(z,t) - \mu_j(z,t)\}/\mu_j(z,t)$.

The average bias is $\delta_j(z) = ED_j(T,z)$ and average relative bias is $\rho_j(z) = ER_j(T,z)$. To see whether they are close to 0, we carried out the following simulation experiment modeled after the diabetes data. Set $\theta = 6$ and $\sigma = \sigma_1 = 0.5$. Let $\mathbf{X} = (X_1, X_2, \ldots, X_{23})$ be the variables listed in Table I, with X_1 and X_2 being HOMA-B and FastBG, respectively (see Figure 5 for their joint distribution). Let $\mathcal{D} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\}$ denote the set of 1077 $\mathbf{X}$ vectors in the diabetes data with nonmissing HOMA-B (172 of the 1249 subjects are missing HOMA-B; none is missing FastBG; Table I). Using $\mathcal{D}$ as the simulation population, randomly draw vectors $\{\mathbf{X}_1^*, \mathbf{X}_2^*, \ldots, \mathbf{X}_n^*\}$ from $\mathcal{D}$ with replacement. For each $\mathbf{X}^* = (X_1^*, X_2^*, \ldots, X_n^*)$, simulate mutually independent $Z^* = 0, 1$ with $P(Z^* = 1) = 0.50$, $b_1 \sim N(0, \sigma_1^2)$, $b_2 \sim U(1.5, 2.5)$, $b_3 \sim U(0.1, 0.2)$, and $\epsilon_j \sim N(0, \sigma^2)$ $(j = 1, 2, \ldots, 9)$. Compute

$$Y_j^* = \theta + b_1 + \frac{b_2 \exp\left(1 - 0.04X_1^*\right)}{1 + \exp\left(1 - 0.04X_1^*\right)} + b_3 \left[4Z^* \left\{ \frac{\exp\left(0.4X_2^* - 4\right)}{1 + \exp\left(0.4X_2^* - 4\right)} - 0.5 \right\} + j - 4.5 \right]^2 + \epsilon_j \tag{4}$$

and its mean conditional on $(\mathbf{X}^*, Z^*)$:

$$\mu_j^*(\mathbf{X}^*, Z^*) = \theta + \frac{2\exp\left(1 - 0.04X_1^*\right)}{1 + \exp\left(1 - 0.04X_1^*\right)} + 0.15 \left[4Z^* \left\{ \frac{\exp\left(0.4X_2^* - 4\right)}{1 + \exp\left(0.4X_2^* - 4\right)} - 0.5 \right\} + j - 4.5 \right]^2.$$

Let T^* denote a tree built from $\mathcal{D}^* = \left\{\left(\mathbf{X}_1^*, \mathbf{Y}_1^*, Z_1^*\right), \left(\mathbf{X}_2^*, \mathbf{Y}_2^*, Z_2^*\right), \ldots, \left(\mathbf{X}_n^*, \mathbf{Y}_n^*, Z_n^*\right)\right\}$, where $\mathbf{Y}^* = (Y_1^*, Y_2^*, \ldots, Y_9^*)$, and let $t^* \in \tilde{T}^*$. Let $\hat{\mu}_j^*(z,t^*)$ denote the sample mean of Y_j^* with $Z^* = z$ in $\mathcal{D}^* \cap t^*$. The population mean of Y_j for $Z = z$ in t^* is $\bar{\mu}_j^*(z,t^*) = |S_t^*|^{-1} \sum_{i\in S_t^*} \mu_j^*(\mathbf{X}_i, z)$, where S_t^* is the set of $\mathbf{X}_i$ in $\mathcal{D} \cap t^*$. The estimation error and relative error are $d_j^*(z,t^*) = \hat{\mu}_j^*(z,t^*) - \bar{\mu}_j^*(z,t^*)$ and $r_j(z,t^*) = d_j^*(z,t^*)/\bar{\mu}_j^*(z,t^*)$, respectively. The corresponding average error and average relative error over $t^* \in \tilde{T}^*$ are $D_j^*(z) = |\tilde{T}^*|^{-1} \sum_{t^*\in\tilde{T}^*} d_j(z,t^*)$ and $R_j^*(z) = |\tilde{T}^*|^{-1} \sum_{t^*\in\tilde{T}^*} r_j(z,t^*)$. Repeating the simulation many times and averaging the values of $D_j^*(z)$ and $R_j^*(z)$ over the number of trials gives estimates of the average bias $\delta_j(z)$ and average relative bias $\rho_j(z)$. Tables III and IV give the results based on 1000 simulation trials for samples of size $n = 100$, 500, and 1000, with m additional independent $U(0,1)$ noise variables, for $m = 0$, 50, and 100 (the number of X_i without noise variables is 23). The bias and relative bias tend to decrease as sample size increases. For $n = 1000$, more than half are within three simulation standard errors of zero. Further, the results seem to be relatively unaffected by the number of noise variables.

4. Bootstrap calibrated intervals

Without confidence intervals to indicate the accuracy of the treatment means, graphs such as that in Figure 1 are not as informative as those in Figures 2–4. Construction of confidence intervals in the terminal nodes of a tree has been a challenging problem, however, since the first regression tree algorithm [23] appeared more than 50 years ago. The difficulty is again due to the terminal nodes being a function of the training data. As a result, the quantities being estimated, such as the node treatment effects, are random. In the context of genome-wide association studies, the problem has been called 'selective

Table III. Estimated average bias $\delta_j(z)$ of treatment means in 1000 simulation trials with standard errors in parentheses; n is sample size and $|\tilde{T}|$ is average number of terminal nodes.

	0 noise variables					
	$n = 100$, $\|\tilde{T}\| = 2.14$		$n = 500$, $\|\tilde{T}\| = 3.58$		$n = 1000$, $\|\tilde{T}\| = 4.34$	
j	$z = 0$	$z = 1$	$z = 0$	$z = 1$	$z = 0$	$z = 1$
1	4e−04 (0.004)	0.036 (0.007)	5e−04 (0.002)	0.010 (0.003)	9e−04 (0.001)	0.006 (0.002)
2	0.015 (0.006)	0.030 (0.007)	0.004 (0.002)	0.010 (0.003)	0.003 (0.002)	0.005 (0.002)
3	0.014 (0.006)	0.033 (0.007)	0.003 (0.002)	0.010 (0.003)	0.001 (0.002)	0.005 (0.002)
4	0.012 (0.006)	0.032 (0.007)	0.003 (0.002)	0.010 (0.003)	0.002 (0.002)	0.005 (0.002)
5	0.012 (0.006)	0.037 (0.007)	0.003 (0.002)	0.011 (0.003)	0.002 (0.002)	0.006 (0.002)
6	0.013 (0.006)	0.042 (0.007)	0.003 (0.002)	0.014 (0.003)	0.002 (0.002)	0.007 (0.002)
7	0.015 (0.006)	0.045 (0.007)	0.005 (0.002)	0.016 (0.003)	0.002 (0.002)	0.007 (0.002)
8	0.017 (0.007)	0.054 (0.008)	0.004 (0.003)	0.020 (0.003)	0.002 (0.002)	0.008 (0.002)
9	0.019 (0.007)	0.064 (0.008)	0.004 (0.003)	0.024 (0.003)	0.001 (0.002)	0.009 (0.002)
	50 noise variables					
	$n = 100$, $\|\tilde{T}\| = 2.01$		$n = 500$, $\|\tilde{T}\| = 3.36$		$n = 1000$, $\|\tilde{T}\| = 4.00$	
j	$z = 0$	$z = 1$	$z = 0$	$z = 1$	$z = 0$	$z = 1$
1	0.008 (0.004)	0.053 (0.007)	0.003 (0.002)	0.011 (0.003)	9e−04 (0.001)	0.008 (0.002)
2	0.025 (0.006)	0.040 (0.007)	0.005 (0.003)	0.010 (0.003)	0.003 (0.002)	0.007 (0.002)
3	0.027 (0.006)	0.048 (0.007)	0.005 (0.002)	0.010 (0.003)	0.003 (0.002)	0.007 (0.002)
4	0.023 (0.006)	0.047 (0.007)	0.004 (0.002)	0.011 (0.003)	0.002 (0.002)	0.008 (0.002)
5	0.025 (0.006)	0.051 (0.007)	0.003 (0.002)	0.013 (0.003)	0.002 (0.002)	0.009 (0.002)
6	0.026 (0.006)	0.055 (0.007)	0.005 (0.002)	0.014 (0.003)	0.003 (0.002)	0.010 (0.002)
7	0.031 (0.006)	0.063 (0.008)	0.005 (0.002)	0.015 (0.003)	0.002 (0.002)	0.011 (0.002)
8	0.027 (0.007)	0.074 (0.008)	0.005 (0.003)	0.018 (0.003)	0.002 (0.002)	0.013 (0.002)
9	0.030 (0.007)	0.082 (0.009)	0.004 (0.003)	0.020 (0.003)	0.003 (0.002)	0.017 (0.002)
	100 noise variables					
	$n = 100$, $\|\tilde{T}\| = 1.98$		$n = 500$, $\|\tilde{T}\| = 3.25$		$n = 1000$, $\|\tilde{T}\| = 3.89$	
j	$z = 0$	$z = 1$	$z = 0$	$z = 1$	$z = 0$	$z = 1$
1	−0.005 (0.004)	0.036 (0.007)	−0.001 (0.002)	0.006 (0.003)	2e−04 (0.001)	0.007 (0.002)
2	0.013 (0.006)	0.036 (0.007)	7e−04 (0.002)	0.005 (0.003)	0.002 (0.002)	0.007 (0.002)
3	0.011 (0.006)	0.037 (0.007)	−1e−04 (0.002)	0.007 (0.003)	0.002 (0.002)	0.007 (0.002)
4	0.008 (0.006)	0.040 (0.007)	0.002 (0.002)	0.006 (0.003)	0.002 (0.002)	0.006 (0.002)
5	0.007 (0.006)	0.046 (0.007)	4e−04 (0.002)	0.005 (0.003)	0.002 (0.002)	0.007 (0.002)
6	0.010 (0.006)	0.048 (0.007)	4e−04 (0.002)	0.008 (0.003)	0.002 (0.002)	0.008 (0.002)
7	0.009 (0.007)	0.056 (0.007)	−9e−04 (0.002)	0.010 (0.003)	0.002 (0.002)	0.011 (0.002)
8	0.012 (0.007)	0.065 (0.008)	7e−04 (0.003)	0.011 (0.003)	0.002 (0.002)	0.013 (0.002)
9	0.011 (0.007)	0.076 (0.009)	−5e−04 (0.003)	0.015 (0.003)	0.002 (0.002)	0.016 (0.002)

inference' [24]. Loh *et al*. [9] describe a method to obtain bootstrap confidence intervals for the case of a single outcome variable. It does so by estimating the standard error of the treatment effects with the standard errors of bootstrap estimates of the corresponding population parameters. Despite simulation results demonstrating its effectiveness, the procedure is unintuitive and lacks justification.

We propose a simpler and more intuitive method based on bootstrap calibration [25], which is briefly explained as follows. Suppose for the moment that there is only one outcome variable, no treatment variable, and t is pre-specified. Given a nominal α, a naïve $100(1-\alpha)\%$ interval for the mean outcome θ_t in t is the t-interval $I(\mathbf{y}, \alpha, t) = \bar{y}_t \pm t_{\nu,\alpha/2} s_t n_t^{-1/2}$. Here, $\mathbf{y}$ denotes a random sample of observations, $\bar{y}_t$, s_t, and n_t its sample mean, standard deviation and sample size in t, and $\nu = n_t - 1$. Let F denote the population from which the data are obtained, and let $\gamma(\alpha, F, t) = P_F\{\theta_t \in I(\mathbf{y}, \alpha, t)\}$ denote the true coverage probability of the interval. Typically, $\gamma(\alpha, F, t) \to 1-\alpha$ as n_t increases if t is fixed. Bootstrap calibration attempts to increase the rate of convergence. Let $\hat{F}$ denote the empirical distribution of $\mathbf{y}$. Using a computer or otherwise, draw bootstrap samples of size n_t from $\hat{F}$. For each bootstrap sample $\mathbf{y}^*$, let $\bar{y}_t^*$ and s_t^* be its sample mean and standard deviation and construct the interval $I(\mathbf{y}^*, \alpha, t) = \bar{y}_t^* \pm t_{\nu,\alpha/2} s_t^* n_t^{-1/2}$. The bootstrap estimate of the coverage probability of $I(\mathbf{y}, \alpha, t)$ is $\gamma(\alpha, \hat{F}, t) = P_{\hat{F}}\{\hat{\theta}_t \in I(\mathbf{y}^*, \alpha, t)\}$, where $\hat{\theta}_t = \bar{y}_t$. Given a target coverage probability $1-\alpha$, find $\hat{\alpha}$ such that $\gamma(\hat{\alpha}, \hat{F}, t) = 1-\alpha$. Then, the $100(1-\alpha)\%$ bootstrap calibrated interval for θ_t is $I(\mathbf{y}, \hat{\alpha}, t) = \bar{y}_t \pm t_{\nu,\hat{\alpha}/2} s_t n_t^{-1/2}$. Loh [25, 26] showed that, under fairly

Table IV. Estimated average relative bias $\rho_j(z)$ of treatment means in 1000 simulation trials with standard errors in parentheses; n is sample size and $|\tilde{T}|$ is average number of terminal nodes.

j	$z=0$	$z=1$	$z=0$	$z=1$	$z=0$	$z=1$
			0 noise variables			
	$n=100$, $\|\tilde{T}\|=2.14$		$n=500$, $\|\tilde{T}\|=3.58$		$n=1000$, $\|\tilde{T}\|=4.34$	
1	−1e−04 (5e−04)	0.005 (8e−04)	0 (2e−04)	0.001 (3e-04)	1e−04 (2e−04)	7e−04 (2e−04)
2	0.002 (7e−04)	0.004 (8e−04)	4e−04 (3e−04)	0.001 (3e-04)	2e−04 (2e−04)	6e−04 (2e−04)
3	0.002 (7e−04)	0.004 (8e−04)	4e−04 (3e−04)	0.001 (3e-04)	1e−04 (2e−04)	7e−04 (2e−04)
4	0.002 (8e−04)	0.004 (8e−04)	4e−04 (3e−04)	0.001 (3e-04)	2e−04 (2e−04)	6e−04 (2e−04)
5	0.002 (8e−04)	0.005 (8e−04)	4e−04 (3e−04)	0.001 (3e-04)	2e−04 (2e−04)	6e−04 (2e−04)
6	0.002 (8e−04)	0.005 (8e−04)	3e−04 (3e−04)	0.002 (3e-04)	2e−04 (2e−04)	7e−04 (2e−04)
7	0.002 (7e−04)	0.005 (8e−04)	5e−04 (3e−04)	0.002 (3e-04)	1e−04 (2e−04)	7e−04 (2e−04)
8	0.002 (7e−04)	0.005 (8e−04)	4e−04 (3e−04)	0.002 (3e-04)	1e−04 (2e−04)	7e−04 (2e−04)
9	0.002 (7e−04)	0.006 (9e−04)	3e−04 (3e−04)	0.002 (3e-04)	1e−04 (2e−04)	8e−04 (2e−04)
			50 noise variables			
	$n=100$, $\|\tilde{T}\|=2.01$		$n=500$, $\|\tilde{T}\|=3.36$		$n=1000$, $\|\tilde{T}\|=4.00$	
1	−7e−04 (5e−04)	0.005 (9e−04)	3e−04 (2e−04)	0.001 (3e−04)	1e−04 (1e−04)	0.001 (2e−04)
2	0.002 (8e−04)	0.005 (9e−04)	6e−04 (3e−04)	0.001 (3e−04)	3e−04 (2e−04)	9e-04 (2e−04)
3	0.002 (8e−04)	0.005 (9e−04)	5e−04 (3e−04)	0.001 (3e−04)	3e−04 (2e−04)	9e-04 (2e−04)
4	0.001 (8e−04)	0.005 (9e−04)	4e−04 (3e−04)	0.001 (3e−04)	3e−04 (2e−04)	0.001 (2e−04)
5	0.001 (8e−04)	0.006 (9e−04)	3e−04 (3e−04)	0.002 (3e−04)	2e−04 (2e−04)	0.001 (2e−04)
6	0.001 (8e−04)	0.006 (9e−04)	5e−04 (3e−04)	0.002 (3e−04)	3e−04 (2e−04)	0.001 (2e−04)
7	0.001 (8e−04)	0.006 (9e−04)	6e−04 (3e−04)	0.002 (3e−04)	2e−04 (2e−04)	0.001 (2e−04)
8	0.001 (8e−04)	0.007 (9e−04)	5e−04 (3e−04)	0.002 (3e−04)	1e−04 (2e−04)	0.001 (2e−04)
9	9e−04 (8e−04)	0.007 (9e−04)	4e−04 (3e−04)	0.002 (3e−04)	3e−04 (2e−04)	0.001 (2e−04)
			100 noise variables			
	$n=100$, $\|\tilde{T}\|=1.98$		$n=500$, $\|\tilde{T}\|=3.25$		$n=1000$, $\|\tilde{T}\|=3.89$	
1	−7e−04 (5e−04)	0.005 (9e−04)	−2e−04 (2e−04)	7e−04 (3e−04)	0 (1e−04)	9e−04 (2e−04)
2	0.002 (8e−04)	0.005 (9e−04)	0 (3e−04)	7e−04 (3e−04)	2e−04 (2e−04)	9e−04 (2e−04)
3	0.002 (8e−04)	0.005 (9e−04)	0 (3e−04)	9e−04 (3e−04)	2e−04 (2e−04)	9e−04 (2e−04)
4	0.001 (8e−04)	0.005 (9e−04)	2e−04 (3e−04)	8e−04 (3e−04)	2e−04 (2e−04)	8e−04 (2e−04)
5	0.001 (8e−04)	0.006 (9e−04)	0 (3e−04)	6e−04 (3e−04)	2e−04 (2e−04)	9e−04 (2e−04)
6	0.001 (8e−04)	0.006 (9e−04)	0 (3e−04)	9e−04 (3e−04)	2e−04 (2e−04)	9e−04 (2e−04)
7	0.001 (8e−04)	0.006 (9e−04)	−2e−04 (3e−04)	9e−04 (3e−04)	2e−04 (2e−04)	0.001 (2e−04)
8	0.001 (8e−04)	0.007 (9e−04)	0 (3e−04)	9e−04 (3e−04)	2e−04 (2e−04)	0.001 (2e−04)
9	9e-04 (8e−04)	0.007 (9e−04)	−1e−04 (3e−04)	0.001 (3e−04)	2e−04 (2e−04)	0.001 (2e−04)

weak conditions, the true coverage probability of $I(\mathbf{y}, \hat{\alpha}, t)$ converges to $1-\alpha$ an order of magnitude faster than the nominal interval $I(\mathbf{y}, \alpha, t)$.

One modification is needed to extend this technique to the nodes of a tree model. Because the tree and its nodes vary from one sample to another, it is not possible to calibrate α for the coverage probability of one particular node. Instead, we calibrate α to improve the average coverage probability over all the nodes of a tree. Let $\bar{\gamma}(\alpha, F) = E\{|\tilde{T}|^{-1} \sum_{t\in\tilde{T}} \gamma(\alpha, F, t)\}$ denote the expected average coverage probability over the terminal nodes of a tree T. Draw bootstrap samples B times from $\hat{F}$ as before to obtain the bootstrap estimate of average coverage probability $\bar{\gamma}(\alpha, \hat{F})$. Do this for a grid of k values $\alpha_1 < \alpha_2 < \ldots < \alpha_k$, with α_k being the desired α level, getting $\bar{\gamma}_i = \bar{\gamma}(\alpha_i, \hat{F})$, $i = 1, 2, \ldots, k$. (We use $k = 5$ in the examples and simulations.) Fit the least-squares line $y = 1 - bx$ to the points $\{(\alpha_1, \bar{\gamma}_1), \ldots, (\alpha_k, \bar{\gamma}_k)\}$ so that $b = \sum_{i=1}^{k} \alpha_i(1-\bar{\gamma}_i) / \sum_{i=1}^{k} \alpha_i^2$. The calibrated $\hat{\alpha}$ is the value such that $\bar{\gamma}(\hat{\alpha}, \hat{F}) = 1 - \alpha$, that is, $\hat{\alpha} = b^{-1}\alpha$, and the bootstrap interval in each node is recalculated with $\hat{\alpha}$ in place of α. The pseudocode in Algorithm 2 finds one $\hat{\alpha}$ for all outcomes by averaging the coverage probabilities over the outcome variables and the terminal nodes. It can be modified to find a separate $\hat{\alpha}$ for each outcome.

To evaluate the accuracy of the bootstrap calibrated intervals, we carried out a simulation experiment using the setup in Section 3.2 as follows. Let $\mathbf{X}^* = \left(X_1^*, X_2^*, \ldots\right)$ denote a random sample drawn with replacement from the diabetes data set $\mathcal{D}$, with X_1 and X_2 being HOMA-B and FastBG, respectively. For each $\mathbf{X}^*$, simulate mutually independent $Z^* = 0, 1$ with $P(Z^* = 1) = 0.50$, $b_1 \sim N\left(0, \sigma_1^2\right)$, $b_2 \sim U(1.5, 2.5)$, $b_3 \sim U(0.1, 0.2)$, and $\epsilon_j^* \sim N(0, \sigma^2)$ $(j = 1, 2, \ldots, 9)$, with $\sigma = \sigma_1 = 0.50$. The simulated jth outcome is

$$Y_j^* = 6 + b_1 + \frac{b_2 \exp\left(1 - 0.04X_1^*\right)}{1 + \exp\left(1 - 0.04X_1^*\right)} + b_3 \left[4Z^* \left\{ \frac{\exp\left(0.4X_2^* - 4\right)}{1 + \exp\left(0.4X_2^* - 4\right)} - 0.5 \right\} + j - 4.5 \right]^2 + \epsilon_j^*$$

and the mean of Y_j^* conditional on $\mathbf{X}^* = (x_1, x_2, \ldots)$ and $Z^* = z$ is

$$\mu_j^*(x_1, x_2, z) = 6 + \frac{2\exp(1 - 0.04x_1)}{1 + \exp(1 - 0.04x_1)} + 0.15 \left[4z \left\{ \frac{\exp(0.4x_2 - 4)}{1 + \exp(0.4x_2 - 4)} - 0.5 \right\} + j - 4.5 \right]^2.$$

Data: Given $\alpha \in (0,1)$, $\alpha_1 < \alpha_2 < \ldots < \alpha_K$; $\mathcal{D} = \{(\mathbf{X}_i, \mathbf{Y}_i, Z_i), i = 1, 2, \ldots, n\}$ with Z_i taking values $1, 2, \ldots, G$; tree with nodes $t_1, t_2, \ldots, t_L$ constructed from $\mathcal{D}$.
Result: $(1 - \alpha)$ confidence interval for $\mu_j(t, z) = E(Y_j \mid t, z)$ for $Z = z$, $t = t_1, t_2, \ldots, t_L$, and $j = 1, 2, \ldots, J$.

```
begin
    γ_k ← 0 for k = 1, 2, ..., K ;          /* bootstrap coverage probabilities */
    for b ← 1 to B do
        bootstrap D*_b = {(X*_i, Y*_i, Z*_i), i = 1, 2, ..., n} from D;
        construct tree from D*_b with nodes t*_b1, t*_b2, ..., t*_bL_b;
        for z ← 1 to G do
            for j ← 1 to J do
                for l ← 1 to L_b do
                    S(t*_bl, z) ← {i | Z_i = z, X_i ∈ t*_bl};
                    a_j(t*_bl, z) ← mean of Y_j ∈ D in S(t*_bl, z);
                    for k ← 1 to K do
                        I_jklz ← nominal-(1 − α_k) interval for μ_j(t*_bl, z);
                        if a_j(t*_bl, z) ∈ I_jklz then
                            c_jklz ← 1 ;              /* interval contains mean */
                        else
                            c_jklz ← 0 ;          /* interval does not contain mean */
                        end
                    end
                end
            end
        end
        for k ← 1 to K do
            γ_k ← γ_k + (GJL_b)^{-1} Σ_j Σ_l Σ_z c_jklz;
        end
    end
    γ_k ← γ_k/B for k = 1, 2, ..., K;
    /* interpolate straight line through fixed point (α, γ) = (0, 1)        */
    s_1 ← Σ_k α_k(1 − γ_k); s_2 ← Σ_k α_k^2; α' ← αs_2/s_1;
    construct nominal (1 − α') intervals for μ_j(t_l, z), l = 1, 2, ..., L;
end
```

Algorithm 2: Bootstrap calibrated intervals

Data: Given $J, K, \sigma^2, \sigma_1^2$, set G of levels of Z, and set
$\mathcal{C} = \{\mathbf{X}_i = (X_{i1}, X_{i2}, \ldots), i = 1, 2, \ldots, n\}$ of design points.
Result: Average coverage probability $p(z)$ of bootstrap intervals for each treatment level z.

begin
  $p(z) \leftarrow 0$, $z \in G$;
  for $k \leftarrow 1$ **to** K **do**
    for $i \leftarrow 1$ **to** n **do**
      Randomly draw $\mathbf{X}_i^* = (X_{i1}^*, X_{i2}^*, \ldots)$ from $\mathcal{C}$;
      Simulate independent Bernoulli Z_i^*, $b_{1i}^* \sim N(0, \sigma_1^2)$, $b_{2i}^* \sim U(1.5, 2.5)$, $b_{3i} \sim U(0.1, 0.2)$, and $\epsilon_{ij}^* \sim N(0, \sigma^2)$;
      Generate $\mathbf{Y}_i^* = (Y_{i1}^*, Y_{i2}^*, \ldots, Y_{iJ}^*)$ from equation (4);
    end
    Fit a tree T^* to $\mathcal{D}^* = \{(\mathbf{X}_i^*, \mathbf{Y}_i^*, Z_i^*), i = 1, 2, \ldots, n\}$;
    Let the terminal nodes of T^* be $t_1, t_2, \ldots, t_{L^*}$;
    Define $\mu_j^*(\mathbf{X}_i, Z_i) = E_{T^*}(Y_{ij} \,|\, \mathbf{X}_i, Z_i)$;
    for $z \in G$ **do**
      $h(z) \leftarrow 0$;
      for $l \leftarrow 1$ **to** L^* **do**
        $\eta_j(t_l, z) \leftarrow |S_l|^{-1} \sum_{i \in S_l} \mu_j^*(\mathbf{X}_i, Z_i = z)$, where $S_l = \mathcal{C} \cap t_l$;
        Use $\mathcal{D}^*$ and Algorithm 2 to find interval $I_{jl}(z)$ for $\eta_j(t_l, z)$;
        for $j \leftarrow 1$ **to** J **do**
          if $\eta_j(t_l, z) \in I_{jl}(z)$ **then**
            $h(z) \leftarrow h(z) + 1$;
          end
        end
      end
    end
    $p(z) \leftarrow p(z) + h(z)/(JL^*)$;
  end
  $p(z) \leftarrow p(z)/K$;
end

Algorithm 3: Estimating coverage of intervals from Algorithm 2

Fit a tree model to each sample $\{(\mathbf{X}_i^*, \mathbf{Y}_i^*, Z_i^*), i = 1, 2, \ldots, n\}$. Then, construct a naïve (i.e., uncalibrated) and a bootstrap calibrated interval for the treatment mean in each node. See Algorithm 3 for the simulation details.

Table V gives the coverage probabilities of the 95% naïve and bootstrap intervals averaged over the nine outcomes for $n = 100$, 500, and 1000, with $m = 0$, 50, and 100 independent $U(0, 1)$ noise variables added. The results are based on 1000 simulation trials with $B = 25$ bootstrap iterations per trial (the real data examples employ $B = 100$). The coverage probabilities of the bootstrap intervals are clearly much closer to the nominal value of 0.95.

5. Mammography screening

Algorithms for subgroup identification that depend on estimation of a treatment-covariate interaction are typically applicable to treatments with two levels only. Our next data set has two outcome variables and a three-level treatment variable. CAPE [27] is a randomized controlled trial designed to determine whether two interventions (DVD or Phone) are more efficacious than a control treatment at promoting

Table V. Coverage probabilities (standard errors in parentheses) of 95% confidence intervals for treatment means averaged over terminal nodes in 1000 simulation trials with 25 bootstrap iterations per trial.

		Naïve intervals		Bootstrap intervals	
m	n	$Z = 0$	$Z = 1$	$Z = 0$	$Z = 1$
0	100	0.907 (0.004)	0.910 (0.003)	0.958 (0.002)	0.961 (0.002)
	500	0.922 (0.003)	0.928 (0.002)	0.946 (0.002)	0.950 (0.002)
	1000	0.928 (0.002)	0.934 (0.002)	0.942 (0.002)	0.947 (0.002)
50	100	0.912 (0.004)	0.917 (0.003)	0.963 (0.002)	0.964 (0.002)
	500	0.923 (0.003)	0.930 (0.002)	0.955 (0.002)	0.956 (0.002)
	1000	0.936 (0.002)	0.939 (0.002)	0.950 (0.002)	0.952 (0.002)
100	100	0.917 (0.004)	0.912 (0.004)	0.967 (0.002)	0.963 (0.002)
	500	0.931 (0.003)	0.933 (0.002)	0.960 (0.002)	0.961 (0.002)
	1000	0.935 (0.002)	0.933 (0.002)	0.949 (0.002)	0.946 (0.002)

m is number of noise variables; n is sample size.

mammography screening (1 = yes, 0 = no) at 6 and 21 months (Resp6 and Resp21, respectively) post-baseline among women 51–75 years old who have not had a mammogram in the previous 15 months. There are 1638 subjects in total, all with Resp6 but 145 without Resp21. Table VI lists the variables and their numbers of missing values.

Logistic regression, applied to each outcome separately on the subjects with complete observations, finds no significant differences between DVD and control, or between phone and control, although there is a significant interaction for the 6-month screening outcome. For women in the low (⩽30K) or middle (30–75K) income categories, DVD is significantly more efficacious than control, and for women in the highest income category (⩾75K), DVD is significantly less efficacious than control. There are also some significant interactions between intervention groups and covariates, such as baseline belief scale scores, for the 21-month outcome. These tests, however, are carried out by testing 2×2 interactions in logistic regression without controlling for multiplicity.

We first analyze the data using the subset of 1493 subjects with both 6-month and 21-month outcomes. Figure 6 shows our regression tree model if the outcome variables are not transformed. The corresponding results with PCA and LDA transformations of the outcome variables at each node are shown in Figures 7 and 8, respectively. Variables `sf12gh` (general health score) and `yearmam` (number of years had a mammogram in past) appear in all three trees. Variable `opt` (optimism scale score) appears in the latter two trees, and fear (perceived fear scale score) appears in the first and third trees. The split points are similar across trees. Mean outcomes are clearly lower if the number of years a subject had a mammogram in the past is 0 or 1 (`yearmam` $\leqslant 1$).

The subgroup {`sf12gh` > 72, `fear` $\leqslant 18$, `yearmam` $\leqslant 1$} in Figure 6 shows statistically significant differential treatment effects. Subjects in the DVD treatment group have the lowest average response rates, and according to the bootstrap confidence intervals, the rates are significantly lower than those for phone. But the rates for phone are not significantly higher than those for control. This subgroup is quite small, however, with 117 subjects.

The results are slightly different in Figure 7, where the subgroup showing statistically significant differential treatment effects is {`opt` > 13, `sf12gh` > 72, `yearmam` $\leqslant 1$}. The DVD treatment group still has the lowest average response rates, and they are significantly lower than those for phone. But the DVD rate at 21 months is also significantly lower than that for control. The subgroup sample size is 264.

The subgroup with statistically significant differential treatment effects in Figure 8 is {`opt` > 13, `sf12gh` > 72, `fear` $\leqslant 21$, `yearmam` $\leqslant 1$}. It is a subset of the subgroup in Figure 7. Here, the response rates for DVD are not significantly lower than those for control, but this may be due to the sample size being smaller at 143.

6. Missing outcomes

Although we have used only subjects with observations in all Y_j variables so far, our method can include subjects who are missing some (but not all) Y_j. Recall that split variable selection at each node is achieved by performing a lack-of-fit test on one Y_j at a time. Therefore, each test can utilize all subjects with nonmissing values in that Y_j, and the same subjects can be used to fit model (1) in the child nodes for

Table VI. CAPE variables and numbers of missing values among the 1493 subjects with complete outcomes and the full set of 1638 subjects

Name	Definition	#Missing	
Resp6	mammography screening 6 months post baseline (yes/no)		
Resp21	mammography screening 21 months post baseline (yes/no)	0	145
Treatment	1 = dvd, 2 = phone, 3 = control		
age	age	1	1
educyrs	years of education		
collegeormore	four-year college or more (1 = yes, 0 = no)		
caucasian	Caucasian (1 = yes, 0 = no)	5	5
afram	African American (1 = yes, 0 = no)	5	5
married	married or in long term relationship (1 = yes, 0 = no)		
income3	household income (1 =< 30K, 2 = 30 − 75K, 3 =>75K)	31	38
incle75k	household income ⩽ 75K (1 = yes, 0 = no)	31	38
workpay	currently working for pay (1 = yes, 0 = no)		
stgpca	baseline stage of behavior change (pre-contemplation/contemplation)		
stage	baseline stage of behavior change (pre-contemplation, contemplation, relapse pre-contemplation, relapse contemplation)		
prepar	baseline preparation (made appointment for mammogram) (1 = yes, 0 = no)		
mediasource	number of 8 media sources exposed to		
paper	exposure to paper media (1 = yes, 0 = no)		
tv	exposure to TV media (1 = yes, 0 = no)		
internet	exposure to internet media (1 = yes, 0 = no)		
hadmamm	ever had a mammogram (1 = yes, 0 = no)		
yearmam	Number of years had a mammogram in past		
doceversug	doctor ever suggest you have a mammogram (1 = yes, 0 = no)		
docspoke	doctor/nurse spoke to you last 2 years about mammogram (1 = yes, 0 = no)		
famhist	family history of breast cancer (1 = yes, 0 = no)		
hcremind	Received reminders of mammogram from health care facility (1 = yes, 0 = no)		
opt	baseline optimism scale score		
sf12bp	baseline SF12 bodily pain scale score		
sf12gh	baseline SF12 general health scale score		
sf12mh	baseline SF12 mental health scale score	1	1
sf12pf	baseline SF12 physical functioning scale score		
sf12re	baseline SF12 role emotional scale score		
sf12rp	baseline SF12 role physical scale score	1	1
sf12sf	baseline SF12 social functioning scale score		
sf12vt	baseline SF12 vitality scale score	0	1
bar	baseline perceived barriers scale score		
ben	baseline perceived benefits scale score		
self	baseline perceived self efficacy scale score		
susc	baseline perceived susceptibility scale score		
fear	baseline perceived fear scale score		
fatal	baseline perceived fatalism scale score		
know	baseline perceived knowledge scale score		

computation of the total sum of squared residuals. This extension can be used with weights as well, but it does not allow PCA and LDA transformations.

An interesting question is whether one should use this more general approach or restrict the models to subjects with complete outcomes. Certainly, restricting to completely observed outcomes makes implicit assumptions about the reasons for the outcomes being missing. If we use all 1249 subjects with at least one outcome in the diabetes data, the pruned tree has no splits. On the other hand, if we use all 1638 subjects with one or more outcomes in the mammography data, we obtain the tree in Figure 9. It is the same as the tree (Figure 6) based on the subset of subjects with complete outcomes. The node treatment means differ slightly between the trees, due to different numbers of observations; compare, for example, the treatment means at the node {`sf12gh` >72, `fear` >18}. Because the model is the same with and without excluding subjects with incomplete outcomes, the result seems to suggest that the outcomes are missing at random. These two examples show that it is useful in practice to analyze the data with and without the subjects with incomplete outcomes.

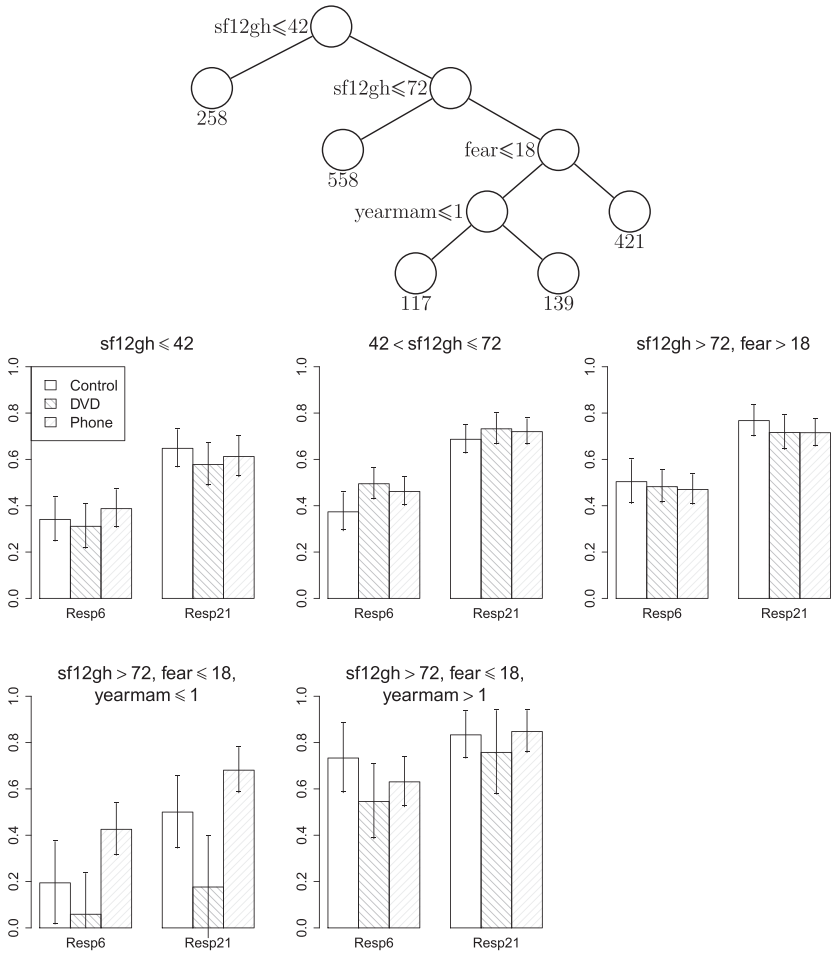

Figure 6. Model based on subset of 1493 subjects without transformations. At each split, an observation goes to the left branch if and only if the condition is satisfied. Error bars are 95% bootstrap confidence intervals. Sample sizes are below terminal nodes.

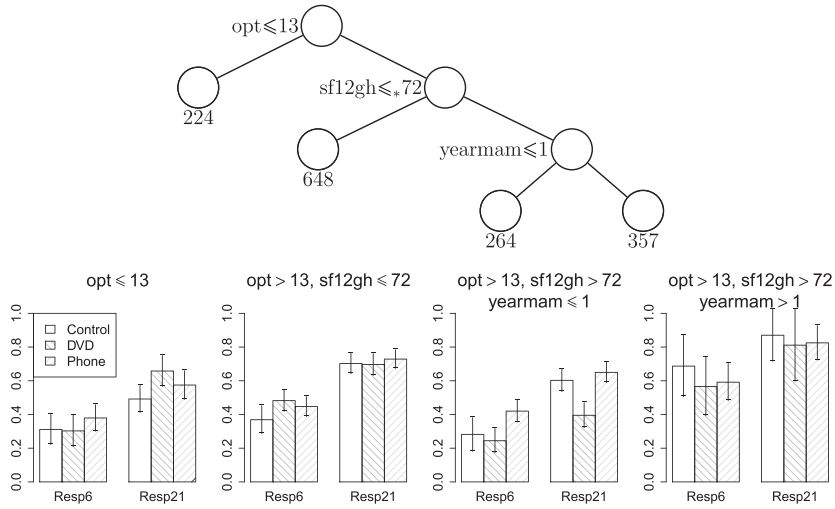

Figure 7. Model based on subset of 1493 subjects with PCA transformations in each node. At each split, an observation goes to the left branch if and only if the condition is satisfied. Error bars are 95% bootstrap confidence intervals. Sample sizes are below terminal nodes.

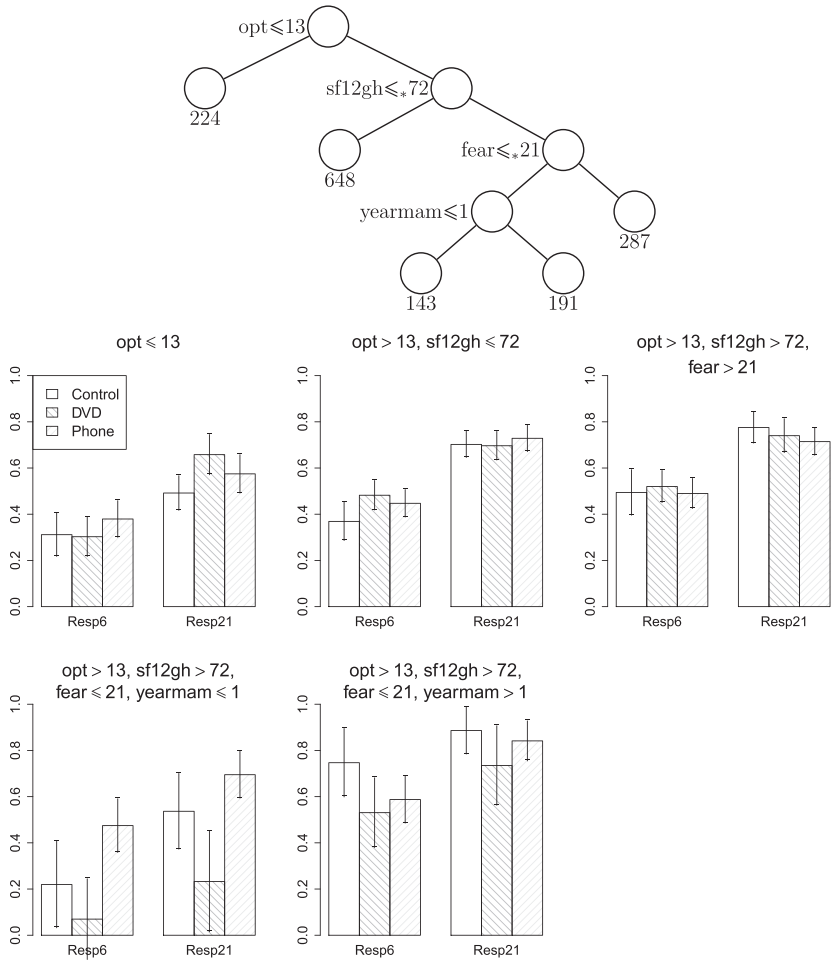

Figure 8. Model based on subset of 1493 subjects with LDA transformations in each node. At each split, an observation goes to the left branch if and only if the condition is satisfied. Error bars are 95% bootstrap confidence intervals. Sample sizes are below terminal nodes.

To our knowledge, the interaction trees approach [3] is currently the only other method that can deal with incomplete outcomes. It uses the CART paradigm, which searches all split points on all split variables at each node. For longitudinal outcomes with binary treatment $Z = 0$ or 1, interaction trees fits a GEE [28] model with mean function

$$EY_j = \beta_0 + \beta_1 I(Z = 1) + \beta_2 I(X_k \in A_k) + \beta_3 I(Z = 1, X_k \in A_k), \quad j = 1, 2, \ldots \tag{5}$$

to the data in each node and each predictor X_k. Here, $A_k = (-\infty, c]$ if X_k is ordinal and is a subset of the levels of X_k if it is categorical. The set A_k that minimizes the p-value of the test that $\beta_3 = 0$ is selected to split the node. One advantage of this approach is that correlations between the Y_j are easily incorporated through specification of a correlation matrix. Another advantage is that subjects can have incomplete outcomes. But it also has disadvantages. The most obvious is the computational cost of fitting many GEE models. An ordinal variable X_k with m unique values generates $(m-1)$ split sets A_k. A categorical variable X_k with m levels generates $(2^{m-1} - 1)$ split sets. Therefore, the number of GEE models to be fitted at each node is linear or exponential in m for each X_k. In practice, specification of the correlation matrix is nontrivial. Su *et al.* [3] use a matrix with constant correlation, but correlations between outcomes far apart in time may be weaker than those nearer together, and the correlations in one child node may differ from those in its sibling node. Further, Equation (5) assumes that the interaction coefficient β_3 is constant over j. This may not be realistic if the number of outcome variables is large. Finally, as with all algorithms that rely on direct optimization, the approach is susceptible to selection bias, because variables that allow more splits have an inherent advantage to be selected.

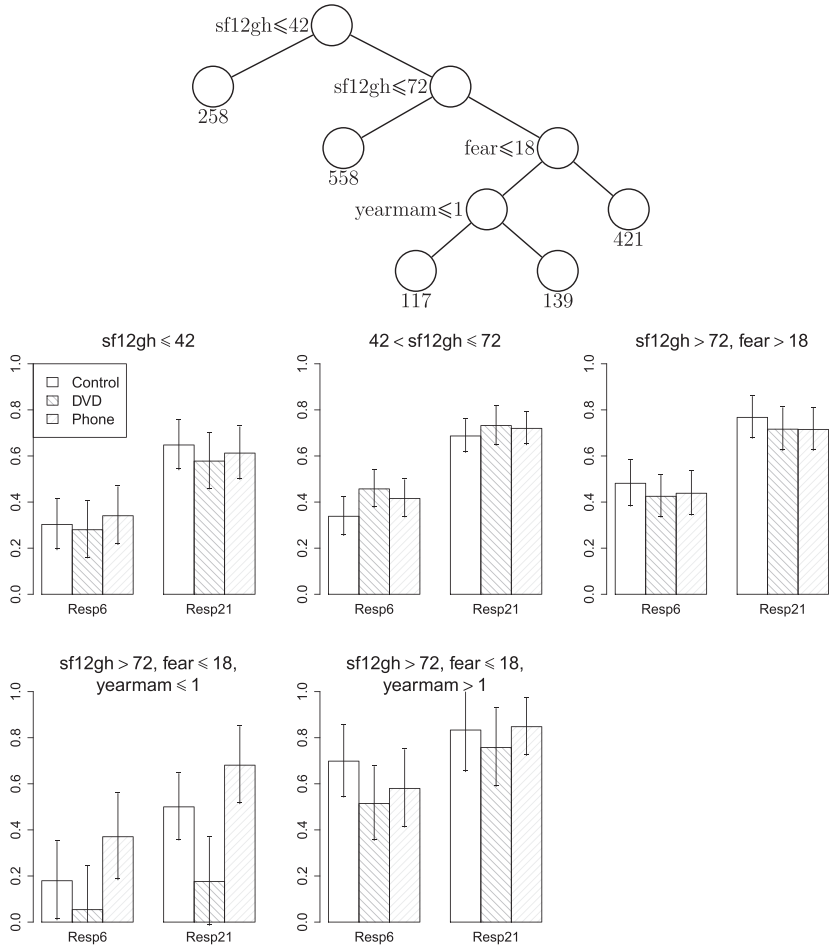

Figure 9. Model based on all 1638 subjects without transformations. At each split, an observation goes to the left branch if and only if the condition is satisfied. Error bars are 95% bootstrap confidence intervals. Sample sizes are below terminal nodes.

7. Concluding remarks

We have described a technique to fit regression tree models that identify subgroups with differential treatment effects from randomized trials with multivariate outcome variables. To our knowledge, it is the first (tree or non-tree) subgroup procedure to accept two or more outcomes simultaneously, to allow missing values in predictor and outcome variables, to allow treatments with more than two levels, and to possess unbiased variable selection and approximately unbiased estimates of treatment effects in the subgroups. Further, if outcomes are completely observed, it can take advantage of local dependencies among outcome variables by employing PCA or LDA transformations at each node of the tree. The examples indicate that either PCA or LDA may be better than no transformation, but the best solution is probably data dependent. In our experience, having more than one solution is often desirable in practice, because they allow the user to apply subject matter knowledge to compare them.

The machine learning approach to subgroup identification typically treats the problem as optimization: search for the subgroups that have the greatest *observed* differential treatment effects. This necessarily produces biased estimates of the effects. Our two-step method escapes this consequence by taking a different tack. The first step ensures that, with high probability, the right variable is chosen to split the node without selection bias. The second step finds the split that fits the data in both subnodes best, without directly maximizing the observed treatment effect in one at the expense of the other. It is natural to wonder if intentionally avoiding the split that maximizes the observed differential treatment effect is the best strategy. The answer is certainly yes, if unbiased estimates are desired. But will this strategy find the correct subgroup? If the treatment effect is a smooth monotone function of a continuous variable X, then, there is no subgroup that is 'correct' without additional qualification. For example, suppose the true

model producing the data is $EY = \beta_0 + f(X)I(Z = 1)$, where $f(x)$ is strictly increasing in x. Then, for any c, the subgroup $\{X > c\}$ has a larger treatment effect than its complement $\{X \leqslant c\}$. Larger values of c yield subgroups with larger treatment effects but they are correspondingly smaller in size. Given that no correct subgroup exists in this case, we are left with two choices: (i) find the subgroup with the maximum observed differential treatment effect and suffer the consequences of biased estimates; or (ii) settle for a subgroup with observed treatment effects that may be less than maximal but that yields approximately unbiased effect estimates. Many methods take the first option; we take the second one here.

There are numerous 'engineering choices' that can potentially affect the performance of our algorithm. One is the two-group conversion of an ordinal variable X into a categorical variable H in the lack-of-fit test. If the sample size at the node is large, this may cause some loss of power in selecting the best X. We could avoid this by converting X into three or more groups, but as sample size decreases with partitioning, the groups will quickly have too few observations. An alternative is to start with a larger number of groups at the top levels of the tree and reduce them at the lower levels. Recall, however, that H already has three groups with dichotomization of X if the latter has missing values. Loss of information from dichotomization is often more than offset by the increase in power from having a missing-value group that admits all observations, especially if missingness is informative.

Our bootstrap calibration approach to confidence interval construction is independent of the subgroup identification algorithm. It can be used with any algorithm and applied to any naïve interval. There is no need to adjust or control for the multiplicity of tests in the search algorithm because all the steps are accounted for in the bootstrap procedure. If calibration is performed on the naïve t-interval, however, it is preferable for the effect estimates to be unbiased.

The tree algorithm discussed here is implemented in the GUIDE software available from www.stat.wisc.edu/~loh/guide.html. It also includes a parallel extension to multiple outcomes of the Gs option proposed in Loh *et al*. [9].

Acknowledgements

The authors are grateful to an associate editor and two referees for helpful comments and suggestions that led to improvements in the paper. WY Loh was partially supported by NSF grant DMS-1305725, NIH grant 1P01CA180945-01, and a grant from Eli Lilly and Company. M Yu was partially supported through a Patient-Centered Outcomes Research Institute (PCORI) Award (ME-1409-21219). The views in this publication are solely the responsibility of the authors and do not necessarily represent the views of the PCORI, its Board of Governors or Methodology Committee.

References

1. Su X, Zhou T, Yan X, Fan J, Yang S. Interaction trees with censored survival data. *International Journal of Biostatistics* 2008; **4**(1). Article 2. DOI:10.2202/1557-4679.1071.
2. Su X, Tsai CL, Wang H, Nickerson DM, Bogong L. Subgroup analysis via recursive partitioning. *Journal of Machine Learning Research* 2009; **10**:141–158.
3. Su X, Meneses K, McNees P, Johnson WO. Interaction trees: exploring the differential effects of an intervention programme for breast cancer survivors. *Journal of the Royal Statistical Society: Series C* 2011; **60**:457–474.
4. Foster JC, Taylor JMG, Ruberg SJ. Subgroup identification from randomized clinical trial data. *Statistics in Medicine* 2011; **30**:2867–2880.
5. Breiman L. Random forests. *Machine Learning* 2001; **45**:5–32.
6. Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and Regression Trees*. Wadsworth: Belmont, California, 1984.
7. 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.
8. Dusseldorp E, VanMechelen I. Qualitative interaction trees: a tool to identify qualitative treatment-subgroup interactions. *Statistics in Medicine* 2014; **33**:219–237.
9. Loh WY, He X, Man M. A regression tree approach to identifying subgroups with differential treatment effects. *Statistics in Medicine* 2015; **34**:1818–1833.
10. Loh WY. Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica* 2002; **12**:361–386.
11. Loh WY. Improving the precision of classification trees. *Annals of Applied Statistics* 2009; **3**:1710–1737.
12. Loh WY, Zheng W. Regression trees for longitudinal and multiresponse data. *Annals of Applied Statistics* 2013; **7**: 495–522.
13. Charbonnel BH, Matthews DR, Schernthaner G, Hanefeld M, Brunetti P. A long-term comparison of pioglitazone and gliclazide in patients with Type 2 diabetes mellitus: a randomized, double-blind, parallel-group comparison trial. *Diabetic Medicine* 2004; **22**:399–405.

14. Weisberg S. *Applied Linear Regression* 2nd ed. Wiley: New York, 1985.
15. Li B, Martin EB. An approximation to the F distribution using the chi-square distribution. *Computational Statistics and Data Analysis* 2002; **40**:21–26.
16. 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.
17. Johnson NL, Kotz S, Balakrishnan N. *Continuous Univariate Distributions*, Vol. 1. Wiley, 1994.
18. Loh WY, Eltinge J, Cho M, Li Y. *Classification and Regression Tree Methods for Incomplete Data from Sample Surveys*. arXiv.org, 2016-03.
19. Gastaldelli A, Miyazaki Y, Mahankali A, Berria R, Pettiti M, Buzzigoli E, Ferrannini E, DeFronzo RA. The effect of Pioglitazone on the liver: role of adiponectin. *Diabetes Care* 2006; **29**:2275–2281.
20. Harris EH. Elevated liver function tests in Type 2 diabetes. *Clinical Diabetes* 2005; **23**:115–119.
21. Loh WY, Shih YS. Split selection methods for classification trees. *Statistica Sinica* 1997; **7**:815–840.
22. Loh WY. Fifty years of classification and regression trees (with discussion). *International Statistical Review* 2014; **34**: 329–370.
23. Morgan JN, Sonquist JA. Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association* 1963; **58**:415–434.
24. Benjamini Y, Heller R, Yekutieli D. Selective inference in complex research. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 2009; **367**(1906):4255–4271.
25. Loh WY. Calibrating confidence coefficients. *Journal of the American Statistical Association* 1987; **82**:155–162.
26. Loh WY. Bootstrap calibration for confidence interval construction and selection. *Statistica Sinica* 1991; **1**:477–491.
27. Champion VL, Rawl SM, Bourff SA, Champion KM, Smith LG, Buchanan AH, Fish LJ, Monahan PO, Stump TE, Springston JK, Gathirua-Mwangi WG, Skinner CS. Randomized trial of DVD, telephone, and usual care for increasing mammography adherence. *Journal of Health Psychology* 2016; **21**(6):916–926.
28. Liang KY, Zeger SL. Longitudinal data analysis using generalized linear models. *Biometrika* 1986; **73**:13–22.