Smooth Sigmoid Surrogate (SSS):
An Alternative to Greedy Search in Recursive Partitioning

Xiaogang Su*
Department of Mathematical Science
University of Texas at El Paso, TX 79968

Joseph Kang, Lei Liu,
Department of Preventive Medicine
Northwestern University, Chicago, IL 60611

Qiong Yang,
Department of Biostatistics, School of Public Health
Boston University, Boston MA 02118

Juanjuan Fan, and Richard A. Levine
Department of Mathematics and Statistics
San Diego State University, San Diego, CA 92182

Abstract
Greedy search or exhaustive search plays a critical role in recursive partitioning and their extensions. We examine an alternative method by replacing the indicator threshold function involved in recursive partitioning with a smooth sigmoid surrogate (SSS) function. In many scenarios, it is important to formulate the problem appropriately, which can change the discrete greedy search for the best cutoff point into a one-dimensional continuous optimization problem. The proposed method can dramatically reduce the computational costs with greedy search. More importantly, it helps address the variable selection bias problem by borrowing statistical inference results available with parametric nonlinear regression models. Both simulations and real-data examples are provided to evaluate and illustrate its usage.

1 Introduction
Recursive partitioning or tree-structured methods (Morgan and Sonquist, 1963 and Breiman et al., 1984), together with their various extensions such as multivariate adaptive regression splines (MARS; Friedman, 1991), bagging (Breiman, 1996), boosting (Freund and Schapire, 1996 and Friedman, 2001), and random forests (Breiman, 2001), have formed an important class of learning tools for data mining and statistical modeling. Greedy search (also called exhaustive search) is the dominating method for identifying the best split or knot at each step of recursive partitioning. In this discrete process, all permissible splits are compared via some criterion and the best one is selected. One
common way for speeding up greedy search is to apply an updating formula to compute the splitting statistic for two consecutive cutoff points on a predictor. Examples can be seen in Segal (1992; Section 2.2), LeBlanc and Crowley (1993), and Su, Tsai, and Wang (2008; Appendix A). Other strategies include splitting on percentiles, sub-sampling, and online incremental learning (see, e.g., Domingos and Hulten, 2000), which, nevertheless, reduces the search space and hence makes no guarantee of success in finding the optimal cutoff point. Despite all these efforts, greedy search can be time-consuming, especially when dealing with continuous variables or categorical variables that have many levels in large data sets.

Another problem that has been considered as inherent with greedy search is the variable selection bias. This refers to the phenomenon that a predictor with more values or levels is more likely to be selected as the splitting variable than a predictor with fewer values or levels. Loh and colleagues have made excellent efforts in addressing this issue with a series of papers, as exemplified in Loh (2002). Their main idea is first select the most important variable and then consider splits on the selected variable only. Another suggestion towards this issue is to derive the distribution of maximally selected test statistics. This approach is exemplified by Miller and Siegmund (1982), Shih and Tsai (2004), and Hothorn and Zeileis (2008).

In this article, we propose to use smooth sigmoid functions as an approximation method for greedy search. For simplicity, we term the method as smooth sigmoid surrogate (SSS). The main idea is to replace the step functions involved in the splitting criterion with smooth sigmoid functions, which results in a smooth surrogate splitting criterion. The best cutoff point for each predictor can be estimated as a parameter in a nonlinear model. Very often the estimation can be appropriately reformulated into a one-dimensional optimization problem that can be quickly solved. We demonstrate that this new method provides comparable performance to greedy search yet with higher speed and improved stability. Furthermore, since the search of best cutoff is cast in a nonlinear regression framework, conventional statistical inference can be exploited to facilitate convenient comparisons across different predictors for finding the best split. We show that this alternative splitting method helps alleviate the variable selection bias problem. Another advantage of this method is that it can be flexibly extended to many other recursive partitioning methods designated for different analytic purposes. In most scenarios, the optimization problem remains one-dimensional.

The reminder of the paper is organized in the following manner. Section 2 introduces the SSS method for regression trees where the response variable is continuous and compares it with greedy search using various numerical studies in terms of identification of the true cutoff point, end-cut preference, and variable selection bias. Decision trees with binary outcomes are treated in Section 3. Section 4 outlines the usage of SSS in several other recursive partitioning based methods. Section 5 ends the article with a brief discussion.

2 Smooth Sigmoid Surrogate for Regression Trees

In regression trees, the data available consist of i.i.d. observations \( \{(y_i, x_i) : i = 1, \ldots, n\} \), where \( y_i \) is the \( i \)-th observed value on the continuous response or target value \( Y \) and \( x_i \in \mathbb{R}^p \) is the associated input vector on predictor variables \( (X_1, \ldots, X_p) \). For the time being, we assume that all predictors are continuous. The scenario for categorical predictors, as well as binary variables, will be discussed later. A split of data is induced by a binary question such as “Is \( X_j \) greater than \( c \)?” for \( X_j \). The greedy search for the best split can be viewed as a two-step process. In the first step, the best way to split each predictor, i.e., the best cutoff point or the best subset, is sought; then, in the second
step, comparison is made across all $p$ predictors.

We first consider the first step, namely, how to determine the best cutoff point for a given predictor $X_j$. For simplicity, we denote $X_j$ as $X$. There are two general approaches of splitting data in the literature, either by minimizing within-node impurity or by maximizing between-node difference. The conventional approach as in CART (Breiman et al., 1984; pp.230–232) splits data by maximizing the reduction in within-node impurity or variation caused by the split. Let $i(t)$ denote the impurity measure for node $t$. In regression trees, $i(t)$ is commonly measured by mean squared error $i(t) = \sum_{i \in t} (y_i - \bar{y}_t)^2 / n_t$. Let $\Delta i = i(t) - \{p(t_L) i(t_L) + p(t_R) i(t_R)\}$ denote the reduction in impurity due to a split of the data, where $t_L$ and $t_R$ denote the two child nodes; $p(t_L)$ is the proportion of observations falling into $t_L$ and analogously for $p(t_R)$. For splits based on variable $X$, the best cutoff point $c^*_j$ achieves the greatest reduction in the sense that $c^*_j = \arg \max_c \Delta i$. The other splitting approach is to maximize the between-node heterogeneity, as exemplified by LeBlanc and Crowley (1993). In many scenarios, these two approaches lead to equivalent splitting criteria. For example, the above criteria with regression trees is obviously equivalent to maximizing a two-sample $F$ test statistic that compares the two child nodes. This approach can also be cast into the following linear model:

$$y_i = \beta_0 + \beta_1 \delta(x_i; c) + \varepsilon_i,$$  \hspace{1cm} (1)

where $\delta(x_i; c) = I\{x_i \geq c\} \triangleq \delta_i$ is the indicator function corresponding to the split and $\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$. The two sample $F$ test is the same as the squared $t$ test for testing $H_0 : \beta_1 = 0$ vs. $H_a : \beta_1 \neq 0$.

Greedy search evaluates all permissible splits and selects the best. This can be slow especially for dealing with big data. It also suffers from other problems including variable selection bias. In this article, we propose a new way of splitting data. The main idea is to approximate the threshold indicator function with a smooth sigmoid surrogate (SSS) function. By appropriately formulating the problem, we demonstrate that this new method provides nearly the same cutoff point as greedy search yet with increased speed and stability. Moreover, it helps alleviate variable selection bias with a simple connection to nonlinear regression models.

### 2.1 Sigmoid Functions

A sigmoid function is a mathematical function that has an ‘S’ shape. Very often, the sigmoid function refers to the special case of the logistic function (also called the expit function):

$$\pi(x) = \pi(x; a) = \left\{ 1 + \exp(-ax) \right\}^{-1} = \frac{\exp(ax)}{1 + \exp(ax)}. \hspace{1cm} (2)$$

Besides the expit function, sigmoid functions include the Gompertz function $f(x) = a \cdot \exp\{b \cdot \exp(c \cdot x)\}$, the ordinary arctangent $\arctan(x)$, the hyperbolic tangent $\tanh(x) = (e^{2x} - 1)/(e^{2x} + 1)$, the probit function, algebraic functions of form $f(x) = x/\sqrt{1 + x^2}$, and many others. In fact, the integral of any smooth, positive, “bump-shaped” function will be sigmoidal. Thus any cumulative distribution function (CDF) is sigmoidal. To approximate the indicator function well, another requirement is that the smooth sigmoid function needs to have range on the unit interval $[0, 1]$; hence rescaling might be needed for some of the above-mentioned functions. The choice of the sigmoid function may not be a big concern; we choose the expit function mainly for its common usage in statistics, e.g., logistic regression, and because many of its properties have been studied, e.g., $d\pi(x; a)/dx = a \pi(x; a) \{1 - \pi(x; a)\}$. 


Figure 1 plots the expit function $\pi\{x; a\}$ for different values of $a = 1, 2, \ldots, 100$. The dark line corresponds to the threshold indicator function $I\{x \leq 0\}$.

Figure 1 plots the expit function with different values of the shape or scale parameter $a = 1, 2, \ldots, 100$. As $a > 0$ increases, $\pi(x)$ provides a better approximation to the indicator function $I\{x \geq 0\}$. Similarly, $\pi(x - c; a)$ approximates $I\{x \geq c\}$. On the other hand, $\pi(x)$ approaches linearity for small $a$. Note that $\pi(x - c; -a) = 1 - \pi(x - c; a)$ and hence $\pi(x - c; a)$ with $a < 0$ approximates $I\{x < c\}$. Either $\pi(x - c; a)$ or $\pi(x - c; -a)$ can be used to represent a split in the recursive partitioning setting. Therefore, our consideration is restricted to $a > 0$ without loss of generality (WLOG). Since trees model data with threshold effects, we are primarily interested in seeking an approximation to the indicator function, where a relatively large $a$ is desirable. For this reason, leaving $a$ for estimation is not advisable. It is clear that the choice of $a$ values also depends on the scale of $x$. For this reason, standardization or other scaling of the predictors is necessary in order to fix $a$ a priori.

### 2.2 Estimating the Best Cutoff Point

A natural approach is to replace $\delta(x_i; c)$ in (1) with an SSS function. The model then becomes

$$y_i = \beta_0 + \beta_1 s(c; x_i) + \varepsilon_i.$$  \hspace{1cm} (3)

where $s(c; x_i) = \pi(x_i - c; a)$. Note that we have suppressed parameter $a$ in the notation since it will be fixed. Model (3) is a nonlinear parametric model, involving four parameters $\{\beta_0, \beta_1, c, \sigma^2\}$. Its estimation and related inference can be done via nonlinear least squares (Seber and Wild, 2003). In fact, the related optimization is a separable least squares problem (Bjorck, 1996; Section 9.4.1).
Estimating Model (3) is a multivariate optimization problem. However, we only need to estimate \( c \) for splitting purposes. This motivates us to approximate the splitting statistic directly and reduce the problem to a one-dimensional optimization with decision variable \( c \) only. Consider the splitting statistic \( \Delta i \) used in CART, which can be treated as an objective function for \( c \) and rewritten as below:

\[
\Delta i(c) = \sum_{i=1}^{n} (y_i - \bar{y})^2 - \left\{ \sum_{i \in L} (y_i - \bar{y}_L)^2 + \sum_{i \in R} (y_i - \bar{y}_R)^2 \right\}
\]

\[
\Delta i(c) \approx \frac{1}{n_L} \cdot \left( \sum_{i \in L} y_i \right)^2 + \frac{1}{n_R} \cdot \left( \sum_{i \in R} y_i \right)^2
\]

\[
\Delta i(c) = \frac{1}{n_L} \cdot \left( \sum_{i \in L} y_i \right)^2 + \frac{1}{n - n_L} \cdot \left( \sum_{i=1}^{n} y_i - \sum_{i \in L} y_i \right)^2 \quad (4)
\]

Throughout the article, we shall use notation \( \cong \) to indicate ‘equivalence or correspondence up to some irrelevant constant’. Namely, we have ignored irrelevant terms that do not involve \( c \) in rewriting \( \Delta i(c) \) in (4). To approximate \( \Delta i(c) \) in (4), one just needs to approximate \( n_L \) and \( \sum_{i \in L} y_i \) by replacing \( \delta(x_i; c) \) with \( s(c; x_i) \) so that

\[
n_L = \sum_{i=1}^{n} \delta(x_i; c) \approx \sum_{i=1}^{n} s(c; x_i)
\]

and

\[
\sum_{i \in L} y_i = \sum_{i=1}^{n} y_i \delta(x_i; c) \approx \sum_{i=1}^{n} y_i s(c; x_i)
\]

The approximated objective function, denoted as \( \tilde{\Delta} i(c) \), becomes

\[
\tilde{\Delta} i(c) = \frac{s^T (yy^T) s}{j^T s} + \frac{(j - s)^T (yy^T) (j - s)}{j^T (j - s)} \quad (5)
\]

where \( y = (y_i), s = \{s_i\} \) with \( s_i = \{s(c; x_i)\} \), and \( j = (1) \) are \( n \)-dimensional vectors.

Suppose further that, WLOG, the response has been centered \( y := (I_n - jj^T/n)y \) so that

\[
\sum_{i=1}^{n} y_i = 0
\]

It follows that \( \sum_{i \in L} y_i = -\sum_{i \in R} y_i \) and hence \( \Delta i(c) \) can be further simplified as

\[
\left( \sum_{i \in L} y_i \right)^2 / \{n_L(n - n_L)\}
\]

up to some irrelevant constant. Approximating \( \Delta i(c) \) leads to the following simple form for \( \tilde{\Delta} i(c) \) :

\[
\tilde{\Delta} i(c) = \frac{s^T (yy^T) s}{s^T (jj^T) (j - s)} \quad (6)
\]

In addition to the cutoff point \( c \), the scale parameter \( a \) is also involved in (5) or (6). Discussion on the effect and choice of \( a \) will follow in the next section. As demonstrated by simulation, the estimated cutoff point \( \hat{c} \) remains essentially steady as long as a large \( a \) value is used. To fix \( a \), it is important to normalize the data \( x_{ij} := (x_{ij} - \bar{x}_j)/\hat{\sigma}_j \), where \( \bar{x}_j, \hat{\sigma}_j \) denote the sample mean and standard deviation of variable \( X_j \), respectively. Alternatively, the range of the data may be normalized into the unit interval \([0, 1]\) by setting

\[
x_{1j} := (x_{ij} - x_{(1)j})/(x_{(n)j} - x_{(1)j})
\]

where \( x_{(1)j} \) and \( x_{(n)j} \) are the minimum and maximum of \( X_j \), respectively. We experimented with both normalization
methods and found no substantial difference in the performance of SSS. We recommend fixing $a = 50$ with normalized data, where the normalization can be done in either way. Once the best cutoff point is identified, we transform it back to the original data scale for interpretability.

With fixed $a$, the best cutoff point $\hat{c}$ can be obtained by maximizing $\Delta i(c)$ in (6) with respect to $c$. The objective function $\Delta i(c)$ is not concave and hence can have many local maxima. Nevertheless, this is a one-dimensional optimization problem. It can be conveniently solved by many standard algorithms. In this paper, we use an algorithm that combines golden section search with successive quadratic approximation, as described in Brent (1973) and implemented in R by the function `optimize()`. This algorithm does not require derivatives, but asks for a range over which the search for the optimum is made. The latter feature allows us to avoid the ‘end-cut preference’ problem (see Section 11.8 of Breiman et al., 1974), which is known as the fact that the use of the least squares criterion tends to favor unbalanced splits, i.e., splits in which one child node has only a very small proportion of observations. It is convenient to define the search interval as $(\xi_\gamma, \xi_{1-\gamma})$, where $\xi_\gamma$ and $\xi_{1-\gamma}$ denote the sample $\gamma$ and $(1-\gamma)$ quantiles of $X$. For all numerical results in this article, we have used $\gamma = 0.02$. 

---

Figure 2: Plot of $\Delta i(c)$ and approximated $\tilde{\Delta} i(c)$ versus permissible cutoff $c$ values. The brown line is $\Delta i(c)$ while the green lines are approximated $\tilde{\Delta} i(c)$ values with $a = 20, \ldots, 100$. 

---

(a) 
(b) 
(c) 
(d) 

(c$_0 = 0.5$) 
(d$_0 = 0.75$)
2.3 Simulation Experiments

To study the performance of the approximation, we simulated data from model $y = 1 + I\{x \leq c_0\} + \varepsilon$, with $x \sim \text{Unif}[0,1]$ and $\varepsilon \sim N(0,1)$. Two sample sizes ($n = 50$ and $n = 500$) and two cutoff points ($c_0 = 0.5$ and $c_0 = 0.75$) are considered, forming four combination scenarios. This allows us to inspect the performance with small versus large samples and balanced versus unbalanced cutoff points. Figure 2 shows the objective function (up to some constant) with each distinct value of $x$ and its approximations with different $a$ values in $\{20, 21, \ldots, 100\}$. It can be seen that large values of $a$ yield similarly good approximations.

Next, we estimated the best cutoff $\hat{c}$ via (6). One hundred simulation runs were taken. Figure 3 plots the estimated cutoff points with different $a$ values for each simulation run. It can be seen that the lines are essentially flat, despite some reasonable minor disturbance. This indicates that estimation of $c$ is fairly steady with respect to the choice of large $a$. Upon standardizing predictors first, we have used $a = 50$ in the subsequent numerical studies.

With the same model settings, Figure 4 presents the empirical density functions (out of 500 simulation runs) of the best cutoff points obtained with SSS, as compared with greedy search. It can be seen that the densities are very similar in each scenario presented. For further numerical comparisons, we also recorded the mean, SD, and median of $\hat{c}$, the mean and median of the bias $|\hat{c} - c_0|$, and the mean squared error $\text{MSE} = \sum_{i=1}^{500}(\hat{c}_i - c_0)^2$ for SSS and greedy search. The results are presented in Table 1. It can be seen that the two methods have quite similar or comparable
Figure 4: Empirical Density of Selected Cutoff Point $\hat{c}$: Smooth Sigmoid Surrogate (SSS; with $a = 50$) vs. Greedy Search. Four scenarios were considered by combining two sample sizes $n \in \{50, 500\}$ and two true cutoff points $c_0 \in \{0.5, 0.75\}$. Each scenario was examined with 500 simulation runs.

performances, indicating that SSS can do as well as GS in finding the underlying cutoff points.

One advantage of SSS over greedy search is computational efficiency. Besides others, two main factors that may affect the computation speed are sample size and the number of distinct values of $X$. To investigate, we generated data from the same model as before, except that $X$ was simulated from a discrete uniform distribution with values in $\{1/k, 2/k, \ldots, k/k\}$ for different values of $k$. We considered different choices of $k$ varying from $\{10, 20, \ldots, 100, 200, 300, \ldots, 10000\}$. The sample size is set at $n = 3k$ so that it increases with $k$. For greedy search, we used the R function `rpart()` by setting its options `rpart.control()` so that it only produces one single split and other features such as competitive and surrogate splits are all suppressed. The implementation of SSS is done solely within R. Figure 5 plots the CPU time (in seconds) that SSS and GS spent on 10 simulation runs. It can be seen that SSS consistently outperforms greedy search, by quite a bit, in all scenarios.

2.4 Other Issues

We note that replacing step functions with smooth sigmoid functions is not a new idea. This tactic has impacted the development of artificial neural networks (ANN). One well-known initial ANN proposal by Rosenblatt (1962) was single-layer networks called perceptrons, which has threshold activation functions. In later developments such as multi-layer perceptrons (MLP), the step function is replaced by the differentiable expit function, which has since advanced ANN tremendously. In
Figure 5: Plot of CPU Computing Time versus the Number (k) of Distinct Values in Variable X for SSS and Greedy Search. The k values considered are \{10, 20, \ldots, 100, 200, 300, \ldots, 10000\}. Variable X was simulated to take values \{1/k, 2/k, \ldots, k/k\} repeatedly three times so that the sample size is n = 3k. The CPU time (in seconds) for 10 simulation runs was recorded.

In the recursive partitioning context, the HME architecture studied by Jordan and Jacobs (1994) is essentially a mixture model where mixing probabilities are hierarchically modelled via multivariate sigmoid or softmax functions. Along similar lines, Ciampi, Couturier, and Li (2002) proposed a ‘soft tree’ procedure where at each node an observation goes to the left or right child node with certain probability that is estimated via soft threshold functions or logistic models. In both procedures, the EM algorithm is a natural choice for model estimation.

We emphasize again that estimating \(\alpha\) is not a good idea for our purpose. If \(\alpha\) is treated as a free parameter, a small \(\alpha\) estimate that corresponds to a nearly linear relationship may be found to provide a better fit. Leaving \(\alpha\) for estimation is important in other model settings such as HME (Jordan and Jacobs, 1994) and soft trees (Ciampi et al., 2002), but does not suit ordinary tree procedures well. This is because tree models account for the relationships between response and predictors via hard threshold functions.

When predictor X is categorical with K levels, a split on X is induced by questions such as “Does X belong to A?” for a categorical predictor X with levels in set C and any subset A \(\subset\) C. In order to apply the SSS method, we can first ‘ordinal’ize its levels by sorting the group means (see CART, Section 9.4). Let \(\bar{y}_k\) denote the sample mean in the \(k\)-th level of X. Arranging \(\{\bar{y}_1, \ldots, \bar{y}_K\}\) in an ascending order, we then assign the \(k\)-th category of X the rank \(r_k\) of \(\bar{y}_k\) and proceed treating X as ordinal. When X is binary, SSS is neither necessary nor available.

Another issue concerns non-concavity. The objective function in (5) or (6), albeit smooth, is not concave, as demonstrated in Figure 2. As a result, the algorithm may get stuck at a local optimum. While the quadratic approximation in R function `optimize()` seems quite effective in avoiding local optima as shown in our numerical studies, global optimization procedures can be helpful. Thanks
Table 1: Summary of the Best Cutoff Points \( c \) Selected by Smooth Sigmoid Surrogate (SSS) and Greedy Search (GS). Results are based on 500 runs.

<table>
<thead>
<tr>
<th>Method</th>
<th>( n = 50 )</th>
<th>( n = 500 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( c_0 = 0.5 )</td>
<td>( c_0 = 0.75 )</td>
</tr>
<tr>
<td>Mean</td>
<td>SSS</td>
<td>0.498651</td>
</tr>
<tr>
<td></td>
<td>GS</td>
<td>0.499340</td>
</tr>
<tr>
<td>SD</td>
<td>SSS</td>
<td>0.109332</td>
</tr>
<tr>
<td></td>
<td>GS</td>
<td>0.138406</td>
</tr>
<tr>
<td>Median</td>
<td>SSS</td>
<td>0.496980</td>
</tr>
<tr>
<td></td>
<td>GS</td>
<td>0.496766</td>
</tr>
<tr>
<td>Mean Bias</td>
<td>SSS</td>
<td>0.076834</td>
</tr>
<tr>
<td></td>
<td>GS</td>
<td>0.085815</td>
</tr>
<tr>
<td>Median Bias</td>
<td>SSS</td>
<td>0.053905</td>
</tr>
<tr>
<td></td>
<td>GS</td>
<td>0.046204</td>
</tr>
<tr>
<td>MSE</td>
<td>SSS</td>
<td>0.011931</td>
</tr>
<tr>
<td></td>
<td>GS</td>
<td>0.019118</td>
</tr>
</tbody>
</table>

Again to the one-dimensional nature of the problem, one simple solution is to divide the search range into several intervals, find the maximum within each, and then identify the highest maximum among them. We have included this step as an option in our implementation, which may occasionally improve the performance, yet slightly slow down the splitting procedure.

### 2.5 Variable Selection Bias

The selection bias problem is deemed inherent in recursive partitioning. Greedy search gives preference to variables that have more distinct levels or values. As a remedy, Loh (2002; GUIDE) viewed the two steps involved in greedy search reversely by first selecting the most important splitting variable \( X^* \) and then determining the best cutoff point for \( X^* \) via greedy search. Besides negligible variable selection bias, the method is advantageous in reducing the computational cost. A similar idea is followed in Zeileis, Hothorn, and Hornik (2008; MOB). However, determination of the most important splitting variable is an equally difficult problem. In GUIDE, residuals from linear models are first grouped. Every predictor is treated or, otherwise, made categorical to form two-way contingency tables with the residual-based grouping. The most important variable is then selected based on Pearson \( \chi^2 \) tests. In MOB, the selection is based on a stability assessment of the estimated coefficients in a linear model. One potential shortcoming of these approaches is that the variable selected via linear models may not be the one with the most important threshold effect for the purpose of partitioning data.

In other approaches, the best cutoff point is viewed as the one that corresponds to maximally selected test statistics, very often, among a number of \( \chi^2 \) statistics. Equivalent to change-point problems (see, e.g., Yao and Davis, 1986), asymptotic inference on maximally selected statistics resorts to standard results in stochastic processes (see, e.g., Miller and Siegmund, 1982; Boulestaix,
idea is that can be efficiently solved by the variable projection method of Golub and Pereyra (1973). The main as minus base-10 logarithm of the resultant \( p \)-value when referring to a \( \chi^2(1) \) distribution. The best split of the data is the one with maximum logworth. Again, the \( p \)-value, together with the associated logworth is computed by referring to the \( \chi^2(1) \) distribution, if the predictor is binary. This is the key of this approach in avoiding multiple comparisons and hence relieving the selection bias problem.
To demonstrate how this new strategy of splitting helps with the selection bias problem, we simulate data from a null model where the response $Y$ has nothing to do with any of the nine predictors $\{X_1, X_2, \ldots, X_9\}$. The numbers of distinct values for $\{X_1, X_2, \ldots, X_9\}$ are set as $\{2, 3, 4, 5, 10, 20, 50, 100, 500\}$ respectively. Two sample sizes $n = 50$ and $n = 500$ were considered. Figure 6 plots the frequency of each variable being selected for splitting data. It can be seen that greedy search suffers severely from selection bias while SSS substantially alleviates the problem.

2.6 Example: 1987 Baseball Salary Data

For illustration, we consider the 1987 baseball salary data, which has been widely analyzed in the literature. The final CART tree, selected with the 0-SE rule, has eight terminal nodes, as plotted in Figure 7(a). Loh (2002) commented that the two splits based on team86 and team87 (highlighted with dashed lines) are hard to interpret and may be attributable to the selection bias of greedy search. The 1-SE rule yields a final tree of four terminal nodes, which is a complete full tree of depth 3. Figure 7(b) plots the final SSS tree with five terminal nodes. This tree structure was determined following the tree methodology by goodness-of-split (LeBlanc and Crowley, 1993), which combines split-complexity pruning with bootstrap-based bias-correction. Of note, the two categorical variables concerning baseball team membership no longer show up in the final SSS tree. Variable yrs is deemed...
Figure 7: Analysis of 1987 Baseball Salary Data. In (a), $S_1$ and $S_2$ represent specific subsets of baseball teams. Within each terminal node is the mean response (log-transformed salary); underneath is the sample size.
important in the GUIDE tree, which can be confirmed via variable importance ranking in random forests as well. It does not show up in the CART tree, but in the SSS tree. We also conducted a 10-fold cross-validation to compare their predictive performance. The mean squared prediction error measures for the SSS tree, 0-SE CART tree, and 1-SE CART tree are 0.1867, 0.1194, and 0.1915, respectively. Thus the final SSS tree is highly competitive in terms of predictive accuracy.

3 SSS for Decision Trees

SSS can be extended to classification or decision trees in a similar manner. Consider data \( \{(y_i, x_i) : i = 1, \ldots, n \} \) with binary response \( y_i \in \{0, 1\} \). Commonly-used node impurities for decision trees include misclassification error rate, Gini index, and entropy. The Gini index corresponds to treating binary \( Y \) as if continuous and applying OLS directly with the linear regression model. If the entropy measure is used, maximizing the reduction in within-node impurity is equivalent to maximizing the LRT for testing \( H_0 : \beta_1 = 0 \) in logistic model:

\[
\logit(\pi_i) = \log \frac{\pi_i}{1-\pi_i} = \beta_0 + \beta_1 \delta(x_i; c),
\]

where \( \pi_i = \Pr(y_i = 1 \mid x_i) \).

3.1 Cutoff Point Identification and Data Partitioning

Same as earlier, replacing the threshold term \( \delta(x_i; c) \) with an SSS term yields a nonlinear logistic model:

\[
\logit(\pi_i) = \beta_0 + \beta_1 s(c; x_i).
\]

To avoid optimization with multiple parameters, it is preferable to first form an objective function by approximating a legitimate splitting statistic and then optimizing it with respect to the cutoff point \( c \) only. This strategy, along a similar line to Neyman (1949), yields much in terms of computational simplicity. For a node \( t \), let \( n_1 = \sum_{i \in t} y_i \) denote the number of observations with \( y = 1 \), \( n_L \) the number of observations in the left child node, and \( n_{L1} \) the number of observations with \( y = 1 \) in the left child node. Up to some constant, the reduction in impurity is \( \Delta_i \equiv -n_L i(t_L) - n_R i(t_R) \). The following proposition expresses \( \Delta_i \) in decision trees in terms of \( n_L \) and \( n_{L1} \) only.

**Proposition 3.1.** Let \( \pi_t \) denote the proportion of observations with \( y = 1 \) in node \( t \). Let \( \Delta_i \) be the reduction in node impurity when node \( t \) is split into \( t_L \) and \( t_R \). With entropy impurity measure \( i(t) = -\pi_t \log \pi_t - (1 - \pi_t) \log(1 - \pi_t) \), we have

\[
\Delta_i \equiv n_L \log n_{L1} + (n_L - n_{L1}) \log(n_L - n_{L1}) + (n_1 - n_{L1}) \log(n_1 - n_{L1}) \\
+ \{(n - n_1) - (n_L - n_{L1})\} \log\{(n - n_1)(n_L - n_{L1})\} \\
- n_L \log n_L - (n - n_L) \log(n - n_L)
\]

up to some constant. With Gini index \( i(t) = \pi_t(1 - \pi_t) \), we have

\[
\Delta_i \equiv -\frac{n_{L1}(n_L - n_{L1})}{n_L} - \frac{(n_1 - n_{L1})\{(n - n_L) - (n_1 - n_{L1})\}}{n - n_L}.
\]
Proposition 3.1 provides the grounds for an efficient way of estimating the best cutoff point \( c \) via SSS. To do so, we need to approximate only \( n_L = \sum_{i=1}^{n} \delta(x_i; c) \) with \( \hat{n}_L = \sum_{i=1}^{n} s_i \) and \( n_{L1} = \sum_{i=1}^{n} y_i \delta(x_i; c) \) with \( \hat{n}_{L1} = \sum_{i=1}^{n} y_i s_i \) in either (10) or (11). The approximated \( \Delta i \) becomes a smooth objective function of \( c \) alone; denote it \( \hat{\Delta} i(c) \). The best cutoff point \( \hat{c} \) can be found as \( \hat{c} = \arg \max_c \hat{\Delta} i(c) \). This is again a one-dimensional optimization and can be efficiently solved with Brent’s (1973) method.

![Figure 8](image_url)

**Figure 8:** Comparing SSS with greedy search in finding the best cutoff point \( c = 0.5 \) with 500 simulation runs. Panels (a) and (c) are the scatterplots (smoothed in such a way that the frequencies of overlapped points are represented in the electromagnetic spectrum) of the cutoff point identified via greedy search versus the cutoff point identified via SSS. Panels (b) and (d) plot the estimated density curves, where the two density curves from SSS and greedy search completely overlap with each other in (d).

SSS is flexible with respect to the general idea of approximation and optimization. The entropy induced splitting statistic corresponds to the LRT \( \chi^2 \). It is obvious that any other two-sample test statistic such as Pearson \( \chi^2 \) can be used in this approach as well. However, the misclassification rate \( i(t) = \min\{\pi(t), 1 - \pi(t)\} \) does not perform as well empirically as entropy or Gini index (Breiman et al., 1984). Besides, its associated impurity reduction is continuous but not smooth. Approximation with SSS would result in a non-smooth objective function, unless additional approximation is made. For this reason, we shall not discuss it further.

An analogous strategy for preventing variable selection bias as in Section 2.5 can be used in decision trees. Estimation of the nonlinear logistic model (9) can be done by combining the iteratively re-weighted least squares (IRWLS) method with the Gauss-Newton plus variable projection method.
discussed earlier. The algorithm is described in Appendix 6. The LRT for testing $H_0 : \beta_1 = 0$ with the nonlinear logistic model (9) can be shown to have the following form,

$$LRT = 2 \sum_{d \in \{L,R\}} \sum_{k=0,1} \hat{n}_{dk} \log \frac{\hat{n}_{dk}}{E_{dk}},$$

with $E_{dk} = \frac{n_k \hat{n}_d}{n}$, which is similar to its counterpart in the linear logistic model except that the $n_{dk}$ count terms involving the split are now replaced with the approximated $\hat{n}_{dk}$. Its statistical significance refers to a $\chi^2(2)$ distribution, except for the scenario with binary predictors where $LRT \sim \chi^2(1)$. Alternatively, Pearson $\chi^2$ of form

$$\sum_{d \in \{L,R\}} \sum_{k=0,1} (\hat{n}_{dk} - E_{dk})^2 / E_{dk}$$

can be used instead. Comparing across predictors, each with its best cutoff point, the split that yields the smallest $p$-value or the largest logworth is selected to partition the data.

### 3.2 Simulation Studies

To illustrate, we simulated data from a logistic model $\Pr\{y = 1\} = \text{expit}(1 + I\{x \leq c_0\})$, with $x \sim \text{Unif}[0,1]$. As before, two sample sizes ($n = 50$ and $n = 500$) and two cutoff points ($c_0 = 0.5$ and $c_0 = 0.75$) are considered. Table 2 presents the mean, SD, and median of $\hat{c}$, the mean and median of the bias, and the mean squared error MSE of the best cutoff points selected via SSS and greedy search, computed from 500 simulation runs. It can be seen that SSS compares favorably to greedy search in every measure with binary outcome. The position of the true cutoff points does not seem to affect the performance much. To gain further insight, Figures 8(a) and 8(c) provide the scatterplots of cutoff points for the two methods for the case of $c = 0.5$. The case for $c = 0.75$ is very similar and hence not graphically presented. Since there are a lot of overlapping points, we have smoothed the plot in such a way that the frequencies of overlapped points are represented in the electromagnetic spectrum. Figure 8(a) shows the end-of-cut preference problem with the greedy search suffers under smaller sample ($n = 50$): even after the remedial measures, there are still a considerable portion of selected cutoff points close to 0 or 1. This is not the case for SSS. A similar pattern is also shown in the estimated density in Figure 8(b). For larger samples ($n = 500$), SSS and greedy search have nearly identical results.

To investigate the selection bias problem, we simulated data from a null logistic model with similar settings as earlier. Namely, each data set contains nine features $\{X_1, X_2, \ldots, X_9\}$ with the possible number of distinct values equal to $\{2, 3, 4, 5, 10, 20, 50, 100, 500\}$ respectively. The bar plots in Figure 9 provides a side-by-side comparison of the selection frequencies from SSS and greedy search. Clearly, SSS largely eliminates the selection bias problem of greedy search.

### 4 Use of SSS in Other Recursive Partitioning Methods

The SSS approximation approach can be used in many other extensions and variants of recursive partitioning procedures. The key idea is again to appropriately formulate the search for the best cutoff point into a one-dimensional optimization problem. In this section, we outline the use and implementation of SSS in a number of tree methods that are developed for different purposes.

#### 4.1 Longitudinal Data

Segal (1992), as well as others, extended tree methods to correlated data. We illustrate the SSS approach with the two-sample Hoteling $T^2$ statistic, which is a natural choice for dealing with
Table 2: Summary for the Best Cutoff Points \( \hat{c} \) Selected by Smooth Sigmoid Surrogate (SSS) and Greedy Search (GS) in the Case of Binary Responses. Results are based on 500 runs.

<table>
<thead>
<tr>
<th>method</th>
<th>( n=50 )</th>
<th>( n=500 )</th>
<th>( n=50 )</th>
<th>( n=500 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( c=0.5 )</td>
<td>( c=0.75 )</td>
<td>( c=0.5 )</td>
<td>( c=0.75 )</td>
</tr>
<tr>
<td>Mean</td>
<td>SSS</td>
<td>0.522499</td>
<td>0.564654</td>
<td>0.495088</td>
</tr>
<tr>
<td></td>
<td>GS</td>
<td>0.554569</td>
<td>0.575064</td>
<td>0.495948</td>
</tr>
<tr>
<td>SD</td>
<td>SSS</td>
<td>0.171994</td>
<td>0.204720</td>
<td>0.096954</td>
</tr>
<tr>
<td></td>
<td>GS</td>
<td>0.240649</td>
<td>0.272286</td>
<td>0.099490</td>
</tr>
<tr>
<td>Median</td>
<td>SSS</td>
<td>0.537842</td>
<td>0.600572</td>
<td>0.498756</td>
</tr>
<tr>
<td></td>
<td>GS</td>
<td>0.554046</td>
<td>0.618786</td>
<td>0.498756</td>
</tr>
<tr>
<td>Mean Bias</td>
<td>SSS</td>
<td>0.140722</td>
<td>0.214540</td>
<td>0.106389</td>
</tr>
<tr>
<td></td>
<td>GS</td>
<td>0.159578</td>
<td>0.248638</td>
<td>0.064748</td>
</tr>
<tr>
<td>Median Bias</td>
<td>SSS</td>
<td>0.117441</td>
<td>0.158724</td>
<td>0.033735</td>
</tr>
<tr>
<td></td>
<td>GS</td>
<td>0.147732</td>
<td>0.190798</td>
<td>0.033735</td>
</tr>
<tr>
<td>MSE</td>
<td>SSS</td>
<td>0.030027</td>
<td>0.076180</td>
<td>0.009405</td>
</tr>
<tr>
<td></td>
<td>GS</td>
<td>0.060774</td>
<td>0.104594</td>
<td>0.009895</td>
</tr>
</tbody>
</table>

multivariate normal data. Suppose that data available are \( \{(y_i, x_i) : i = 1, \ldots, n\} \), where \( y_i \in \mathbb{R}^q \) is the response vector and \( x_i \in \mathbb{R}^p \) is the predictor vector. Consider splits induced by variable \( X_j \) or, simply, \( X \), for simplicity. To apply SSS, define \( \hat{n}_L = \sum_{i=1}^n s_i \) and \( \hat{m}_L = \sum_{i=1}^n s_i y_i \). An approximated Hoteling \( T^2 \) statistic is given by

\[
\hat{T}^2(c) = \frac{\hat{n}_L(n - \hat{n}_L)}{n} \hat{d}^T S^{-1} \hat{d},
\]

where

\[
\hat{d} = \frac{\hat{m}_L}{\hat{n}_L} - \frac{\sum_{i=1}^n y_i - \hat{m}_L}{n - \hat{n}_L},
\]

\[
S = \sum_{i=1}^n y_i y_i^T - \frac{\hat{m}_L \hat{m}_L^T}{\hat{n}_L} - \frac{\left( \sum_{i=1}^n y_i - \hat{m}_L \right) \left( y_i - \hat{m}_L \right)^T}{n - \hat{n}_L}.
\]

As before, \( \hat{T}^2(c) \) involves \( c \) only and hence can be used as an objective function for one-dimensional optimization. Several other splitting statistics are outlined in Segal (1992), depending on the dependence structure and other specific features of the data. Similar to Hoteling \( T^2 \), they can be derived from either a linear mixed-effects model or the generalized estimating equation (GEE) approach. Each can be combined with SSS for a better way of splitting.

### 4.2 Survival Trees

Consider censored survival data that consists of \( \{(T_i, \Delta_i, z_i) : i = 1, \ldots, n\} \), where the observed death time is \( T_i = \min(T_i', C_i') \) with \( (T_i', C_i') \) denoting the death and censoring times for the \( i^{th} \) individual;
Figure 9: Comparing SSS versus greedy search in selection bias. Bar plots are based on frequencies (from 1,000 simulation runs) of splitting. Variables selected by either method. The data were generated from a null model. Two sample sizes $n = 50$ and $n = 500$ were considered.

$\Delta_i = I\{T_i^r \leq C_i^r\}$ is the survival status indicator; and $z_i = (z_{ij})_{j=1}^p \in \mathbb{R}^p$ is the covariate vector associated with subject $i$. For concerns over identifiability in the ensuing modelling and inference, we assume that $T_i^r$ and $C_i^r$ are independent given $z_i$.

Segal (1992) and LeBlanc and Crowley (1993) extended regression trees to survival data, termed as ‘survival trees’, using the logrank test statistic. We first consider finding the best cutoff point for one variable $Z_j$, or denoted as $Z$ for simplicity. Let $t_1 < t_2 < \cdots < t_D$ be distinct uncensored death times observed in the node $t$. At each $t_k$ for $k = 1, \ldots, D$, a $2 \times 2$ contingency table is induced by a binary split $I\{z_i \leq c\}$ and the survival status as tabulated below, where $Y_k = \sum_{i=1}^n I\{T_i \geq t_k\}$ is the total number of subjects at risk at time $t_k$; $d_k$ is the total number of deaths at $t_k$; $Y_{kL}$ is the number of subjects at risk in the left child node; and $d_{kL}$ is the number of deaths in the left child node. Other entries can be expressed in terms of $(Y_k, d_k, Y_{kL}, d_{kL})$. 

\begin{align*}
Y_k &= \sum_{i=1}^n I\{T_i \geq t_k\} \\
d_k &= \sum_{i=1}^n I\{T_i = t_k\} \\
Y_{kL} &= \sum_{i=1}^n I\{T_i \geq t_k, T_i < t_k\} \\
d_{kL} &= \sum_{i=1}^n I\{T_i = t_k, T_i < t_k\}
\end{align*}
Given \((Y_k, d_k, Y_{kL})\), \(d_{kL}\) follows a hypergeometric distribution with mean \(E_{kL} = Y_{kL}d_k/Y_k\) and variance \(V_{kL} = d_k(Y_k - d_k)Y_{kL}(Y_k - Y_{kL})/\{Y_k^2(Y_k - 1)\}\), under the null hypothesis of no difference in survival between two child nodes. In a similar vein to the Mantel–Haenszel test, the logrank statistic takes the following general form

\[
Q(c) = \left\{ \frac{\sum_{k=1}^{D} w_k(d_{kL} - E_{kL})}{\sum_{k=1}^{D} w_k^2 V_{kL}} \right\}^2,
\]

where \(w_k\) is some additional weight associated with \(t_k\). To apply SSS, note that \((d_{kL}, Y_{kL})\) are the only terms in \(Q(c)\) that involves the cutoff point \(c\). So we just need to approximate them with \((\tilde{d}_{kL}, \tilde{Y}_{kL})\) in a similar way:

\[
\begin{align*}
Y_{kL} &= \sum_{i=1}^{n} I\{z_i \leq c\} I\{T_i \geq t_k\} \quad \rightarrow \quad \tilde{Y}_{kL} = \sum_{i=1}^{n} \pi\{a(z_i - c)\} I\{T_i \geq t_k\} \\
d_{kL} &= \sum_{i=1}^{n} \Delta_i I\{z_i \leq c\} I\{T_i \geq t_k\} \quad \rightarrow \quad \tilde{d}_{kL} = \sum_{i=1}^{n} \Delta_i \pi\{a(z_i - c)\} I\{T_i \geq t_k\}.
\end{align*}
\]

The approximated logrank statistic \(\tilde{Q}(c)\) forms a smooth objective function of \(c\) for one-dimensional optimization. This allows for a fast identification of the best cutoff point for each covariate. Under the condition of no tied death time, it can be shown that the logrank statistic corresponds to the score test of \(H_0: \beta = 0\) in the Cox proportional hazards (PH) model (Cox, 1972)

\[
\lambda_i(t|z) = \lambda_0(t) \exp(\beta I\{z_i \leq c\})
\]

where \(\lambda_i(t|z)\) is the hazard function for subject \(i\) and \(\lambda_0(t)\) is the baseline hazard. This connection provides some leeway to address the selection bias issue in a similar manner as before.

### 4.3 Tree-Augmented Regression (TAR)

Su, Tsai, and Wang (2008) introduced tree-augmented regression (TAR) analysis, in which a tree structure is appended to a ‘best’ linear model for model diagnostics and improvement purposes. Observing that linear regression and tree models complement each other very well in many ways, the idea is to borrow strength from both. Suppose that model

\[
y = X\beta + \varepsilon
\]

is the ‘best’ linear model obtained from some selection procedure. A TAR model can be written as \(y = X\beta + T\gamma + \varepsilon\), where \(T\) denotes a dummy vector induced by an added tree structure. If a non-null tree structure \(T\) can be developed, it indicates lack-of-fit of the ‘best’ linear model and the TAR model tends to perform better. Simonoff (2013) has recently revisited this method with longitudinal data and found that it has excellent empirical size and power in goodness-of-fit assessment.
To fit the TAR model, one quick immediate way is simply run CART on the residuals (Miller, 1996) obtained from Model (12). Su, Tsai, and Wang (2001) considered a different way of splitting and demonstrated its substantial superiority to the residual-based approach. At each node, TAR splits data by considering model
\[ y = X\beta + \gamma \delta + \epsilon, \tag{13} \]
where again \( \delta = (\delta_i) \) with \( \delta_i = I\{x_i \leq c\} \) for predictor \( X \). The initial implementation of TAR is based on an updating formula using QR decomposition and Householder one-step transformation.

To apply SSS, we shall take a closer look at the splitting statistic, which is essentially the least squares criterion for Model (13). Using same space and projection notations introduced earlier, the LS criterion is
\[ Q(c) = \| y - PC_{(X, \delta)} y \|^2, \]
where \( C(X, \delta) \) denotes the column space of matrix \((X, \delta)\). The best cutoff point \( \hat{c} \) is obtained by minimizing \( Q(c) \).

Denote the residual vector of the ‘best’ linear model (12) as \( r_y = y - PC_{(X)} y = PC_{(X)} y \).
Similarly, the residual vector obtained by regressing \( \delta \) on \( X \) is \( r_{\delta} = PC_{(X)} \delta \).
This step of taking the residual is termed ‘deflation’ in partial least squares methods. The following proposition simplifies \( Q(c) \) and compares it to the residual-based approach.

**Proposition 4.1.** Up to some irrelevant constant, \( Q(c) \approx -\|PC_{(r_y)} y \|^2 = - \frac{(y^T r_y)^2}{\|r_y \|^2} = - \frac{\|r_y \|^4}{\delta^T P_{C^\perp(X)} \delta} \).

For the LS criterion based on residuals \( r_y^T \),
\[ \|PC_{(r_y^T)} r_y \|^2 \approx - \frac{(r_y^T \delta)^2}{\delta^T P_{C^\perp(j)} \delta}, \]
where \( j = (1) \in \mathbb{R}^n \) is the n-dimensional vector with components all being 1.

With the above forms, it is interesting to see that the splitting statistic in the TAR approach differs from that in the residual-based approach only with the denominator, i.e., \( P_{C^\perp(X)} \) vs. \( P_{C^\perp(j)} \) for the quadratic form of \( \delta \). Since \( j \) is the first column of \( X \) if model (12) contains an intercept term, \( C^\perp(X) \subseteq C^\perp(j) \) and hence \( \delta^T P_{C^\perp(j)} \delta \geq \delta^T P_{C^\perp(X)} \delta \). Thus, the residual-based approach gives less weight to the sum of cross-products between residuals \( r_y \) and split indicator vector \( \delta \).

To apply SSS, we replace \( \delta \) with \( s \) in \( Q(c) \) and optimize it with respect to \( c \). Namely, the best cutoff point \( c \) solves
\[ \max_c \frac{s^T (r_y r_y^T) s}{s^T P_{C^\perp(X)} s}, \tag{14} \]
which is again a one-dimensional smooth optimization problem. The objective function in (14) has the form of the Rayleigh quotient of matrix \((r_y r_y^T)\) relative to matrix \( P_{C^\perp(X)} \). Both matrices are semi-positive definite and neither involves \( c \). Note that the analytic solution of \( s \) to this Rayleigh quotient, albeit available, may not provide much further help for our purposes since \( s \) has additional specification by the single decision variable \( c \).
4.4 Interaction Trees (IT)

Su et al. (2009) applied recursive partitioning to subgroup analysis by explicitly seeking treatment-by-covariate interactions. Consider data \( \{(y_i, T_i, x_i) : i = 1, \ldots, n\} \) that come from a randomized trial, where \( y_i \) is the \( i \)th continuous response; \( T_i \) is the binary treatment assignment indicator: 1 for the treated group and 0 for control; and \( x_i \in \mathbb{R}^p \) is a \( p \)-dimensional covariate vector. Given a split, the following 2 \( \times \) 2 table is induced.

<table>
<thead>
<tr>
<th>Treatment</th>
<th>Child Node</th>
<th>Left</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>((\bar{y}<em>{0L}, n</em>{0L}))</td>
<td>((\bar{y}<em>{0R}, n</em>{0R}))</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>((\bar{y}<em>{1L}, n</em>{1L}))</td>
<td>((\bar{y}<em>{1R}, n</em>{1R}))</td>
<td></td>
</tr>
</tbody>
</table>

The splitting statistic in IT is the squared \( t \) test for \( H_0 : \beta_3 = 0 \) in the interaction model:

\[
y_i = \beta_0 + \beta_1 T_i + \beta_2 \delta_i + \beta_3 T_i \cdot \delta_i + \varepsilon_i \text{ with } \varepsilon_i \overset{IID}{\sim} N(0, \sigma^2),
\]

which assesses differential treatment effects between two child nodes. It amounts to

\[
Q(c) = \frac{\{(\bar{y}_{1L} - \bar{y}_{0L}) - (\bar{y}_{1R} - \bar{y}_{0R})\}^2}{\hat{\sigma}^2(1/n_{1L} + 1/n_{0L} + 1/n_{1R} + 1/n_{0R})},
\]

where \( \hat{\sigma}^2 \) is the pooled estimator of \( \sigma^2 \). The best cutoff point is \( c^* = \arg \max_c t^2(c) \). It is worth noting that minimizing the least square criterion with Model (15) is not directly helpful for the purposes of IT. A cutoff point can yield minimum LS merely for its strong additive effect (i.e., associated with \( \beta_2 \)).

To apply SSS, we approximate \( Q(c) \) by starting with \( n_{lt} \) and \( \bar{y}_{lt} \) for \( l = 0, 1 \) and \( t = \{L, R\} \):

\[
n_{lt} = \sum_{i=1}^{n} T_i^l (1 - T_i)^{1-t} \delta_i^{I(t=L)} (1 - \delta_i)^{I(t=R)} \approx \sum_{i=1}^{n} T_i^l (1 - T_i)^{1-t} s_i^{I(t=L)} (1 - s_i)^{I(t=R)} = n_{lt}
\]

\[
\bar{y}_{lt} = \frac{\sum_{i=1}^{n} y_i T_i^l (1 - T_i)^{1-t} \delta_i^{I(t=L)} (1 - \delta_i)^{I(t=R)}}{n_{lt}} \approx \frac{\sum_{i=1}^{n} y_i T_i^l (1 - T_i)^{1-t} s_i^{I(t=L)} (1 - s_i)^{I(t=R)}}{n_{lt}} = \bar{y}_{lt}
\]

Bringing the approximated quantities \((\bar{n}_{lt}, \bar{y}_{lt})\) into

\[
\hat{\sigma}^2 = \frac{1}{n - 4} \left( \sum_{i=1}^{n} \bar{y}_{l}^2 - \sum_{k=0,1} \sum_{t \in \{L, R\}} n_{kt} \bar{y}_{kt}^2 \right)
\]

yields an approximation of \( \hat{\sigma}^2 \). The approximated \( \hat{Q}(c) \) is a smooth objective function of \( c \) only and can be directly maximized for finding the best cutoff point \( \hat{c} \). Similar methods can be used to construct causal inference trees with observational data (Su et al., 2012).

4.5 Multivariate Adaptive Regression Splines (MARS)

Friedman (1991) proposed multivariate adaptive regression splines (MARS), which combines recursive partitioning with regression splines to fit piecewise linear functions. The spline basis functions used are the truncated polynomials. To maintain computational feasibility, MARS considers only
the first-order terms, i.e., a basis pair of form \((x_i-c)_+\) and \((x_i-c)_-\) for predictor \(X_j\) (with \(j\) omitted again for simplicity), where \(x_+ = x \lor 0\) and \(x_- = (-x) \lor 0\) denotes the positive and negative part of \(x\), respectively. Since \(x_- = x_+ - x\), the two terms are equivalent to

\[ x_i \quad \text{and} \quad (x_i - c)_-, \tag{17} \]

where the constant \(c\) has been absorbed by the intercept term.

Starting with the intercept, MARS repeatedly expands the model by seeking the best-performing basis pair that interact with one of the terms in the current model. Suppose that the current model

\[ y = \beta_0 b_0 + \beta_1 b_1 + \cdots + \beta_k b_k + \varepsilon, \]

contains \((k + 1)\) basis terms \(\{b_l\}_{l=0}^k\). In particular, \(b_0 = \mathbf{j} = (1)\) corresponds to the intercept. Let \(D_l = \text{diag}(b_l)\) be a \(n \times n\) diagonal matrix with \(b_l\) as diagonal elements for some \(l \in \{0, 1, 2, \ldots, k\}\). Denote \(b_x = (x_i)\) and \(b_x(c) = ((x_i - c)_-)\). The model in the next step is given by

\[ y = \beta_0 b_0 + \beta_1 b_1 + \cdots + \beta_k b_k + \gamma_1 D_l b_x + \gamma_2 D_l b_x(c) + \varepsilon. \tag{18} \]

Note that elements in \(D_l b_x\) are cross-product terms \(x_i b_l\); similarly for \(D_l b_x(c) = \{(x_i - c)_- b_l\}\). When \(l = 0\), these are additive terms from the basis pair. When \(l \geq 1\), they become cross-product interaction terms. For the sake of computational efficiency, MARS poses more constraints, e.g., only first-order interaction between two different predictors are considered. The reader is referred to Friedman (1991) for more details.

To develop model (18), one needs to determine the best knot \(\hat{c}\) for every predictor \(X_j\) and for each \(l \in \{0, 1, 2, \ldots, k\}\). Comparing across all these choices, the best basis pair is then selected. It is quite time consuming to find the best knot \(\hat{c}\). To speed up the learning, an updating formula for consecutive values of \(X_j\) is developed in MARS. We next show that SSS can be used here to facilitate a quick identification of \(\hat{c}\).

To illustrate, we fix \(X_j\) (simply as \(X\)), as well as the basis function \(b_l\), and seek the best knot \(\hat{c}\). First note that the spline basis pair in (17) can be written as

\[ x_i \quad \text{and} \quad (x_i - c) \cdot I(x_i \leq c) = (x_i - c) s_i(c). \]

Only the latter term \(b_i\) involves \(c\), which can be approximated with

\[ (x_i - c) \cdot \pi \{a(x_i - c)\} = (x_i - c) s_i(c). \]

Similarly for Model (18), \(c\) is only involved in the term \(\gamma_2 D_l b_x(c)\). Now we are in the same scenario as that involved in TAR. Let \(B = (b_1, \ldots, b_k, D_l b_x)\) and \(r = P_{C^\perp(B)} y\). Let \(\tilde{d}_x(c) = \{(x_i - c)_- s_i(c)\}\). Referring to Section 4.3, the best knot \(\hat{c}\) can then be found as

\[ \hat{c} = \arg\max_c \frac{\tilde{d}_x(c)^T (D_l r r^T D_l) \tilde{d}_x(c)}{\tilde{d}_x(c)^T (D_l P_{C^\perp(B)} D_l) \tilde{d}_x(c)}, \]

where the objective function is again the Rayleigh quotient of quadratic forms with respect to two semi-positive definite matrices. Besides, variable selection bias, although rarely mentioned in the literature, is also a problem for MARS. With the SSS approach, one can better address this deficiency by again referring to a nonlinear parametric model, in which the best knot consumes one additional degree of freedom.
5 Discussion

We have explored a simple and intuitive alternative method to greedy search in tree-structured methods. The initial motivation of SSS came from the use of smooth sigmoid functions in artificial neural networks, HME (Jordan and Jacobs, 1994), soft trees (Ciaimi, Couturier, and Li, 2002), etc. Nevertheless, having a relatively large scale parameter \( a \) forces a hard threshold, which can serve better as an approximation in partitioning-based methods. The smoothness induced by SSS has several appealing consequences. These include easier optimization, higher computational speed, and certain immunity to the end-cut preference problem. Moreover, its natural connection with nonlinear regression renders a way for alleviating the selection bias problem. From Figures 6 and 9, it can be seen that selection bias remains with SSS to some extent, i.e., smaller selection proportions for variables with smaller numbers of distinct values or levels. As potential future research, we think that a Bartlett (1937; 1954) correction to LRT could further improve the results. In particular, one closely related reference to SSS in regression trees is Cordeiro (2004).

We have confined attention to the use of SSS only in settings where the splitting statistic is of explicit form. This allows us to formulate identification of the best cutoff point as a one-dimensional smooth optimization problem, for which many algorithms are applicable. We have chosen the Brent (1973) method that combines parabola interpolation with golden section search for its easy accessibility and super performance; its range constraint feature also allows us to better handle the end-cut preference problem. Our results suggest that implementation of many tree-based methods and their extensions can benefit from SSS. For example, the use of SSS in random forests (Breiman, 2001) may lead to improved variable importance ranking. In other scenarios where we do not have a closed-form for the splitting criterion, SSS can be useful as well. For example, the extension of interaction trees to longitudinal data (Su et al., 2011) involves estimating marginal models with interaction and threshold terms for all permissible splits. Using SSS can greatly reduce the computational burden. The resultant nonlinear model can be estimated with the aid of the separable least squares as discussed in Appendix 6.

Acknowledgement

The work was partially done while Xiaogang Su was visiting the Institute for Mathematical Sciences, National University of Singapore in 2014. The visit was supported by the Institute.

APPENDIX

6 Gauss-Newton Algorithm with Variable Projection

To estimate the nonlinear model (3): \( y_i = \beta_0 + \beta_1 s(c; x_i) + \epsilon_i = \eta_i + \epsilon_i \), the Gauss-Newton algorithm with variable projection (Kaufman, 1975) is described in Algorithm 1. In the algorithm, note that the intermediary updating step on \( \beta \) is actually

\[
\beta := \beta + \delta = (S^T S)^{-1} S^T y.
\]
Denote $\eta = S\beta = [\eta_i]$. The Jacobian matrix of $\eta$, denoted as $F$, is 

$$F = \frac{d\eta}{d\theta} = [S, v], \quad \text{with } v = \frac{d\eta}{dc} = [-\alpha \beta_1 s_i (1 - s_i)].$$

Matrix $F$ is closely related to the linearization step in nonlinear models and plays a critical role in both the optimization algorithm and statistical inference.

**Algorithm 1** Gauss-Newton Algorithm with Variable Projection

- **initialize** $\theta = (\beta_0, \beta_1, c)^T$.
  
- **repeat**
  
  - compute $\delta = (S^T S)^{-1} S^T r$ by first obtaining $S$ and residual vector $r = y - S\beta$;
  
  - update $\beta := \beta + \delta$.

  - evaluate Jacobian $F = d\eta/d\theta$ at current $(\beta, c)$;

  - compute direction $d = (F^T F)^{-1} F^T (y - S\beta)$;

  - update $\theta := \theta - \alpha d$ with step size $\alpha > 0$;

- **until** convergence.

We next consider estimation of the nonlinear logit model (9): $\log \{ \pi_i / (1 - \pi_i) \} = \beta_0 + \beta_1 s(c; x_i)$.

The associated log-likelihood function is 

$$\ell(\theta) = \sum_{i=1}^{n} \{ y_i \eta_i + \log(1 - \pi_i) \}.$$

Note that $\theta$ enters the likelihood through $\eta_i$ and $\pi_i = \expit(\eta_i)$. Fisher scoring, a popular variant of Newton-Raphson where the Hessian matrix is replaced by its expected value for simpler form, is the standard method for GLM. To apply, it can be verified that the gradient is 

$$g = \frac{d\ell}{d\theta} = F^T(y - \pi)$$

and the minus expected Hessian matrix (which is $n$ times Fisher information matrix) is $F^T W F$, where matrix $W$ is diagonal with diagonal elements $w_i = \pi_i (1 - \pi)$ for $i = 1, \ldots, n$. It is worth noting that the observed Fisher information matrix is not equal to the expected Fisher information matrix with this nonlinear logistic model, even though the canonical logistic link is used.

Therefore, the updating step is 

$$\theta := \theta + (F^T W F)^{-1} F^T (y - \pi)$$

$$:= (F^T W F)^{-1} F^T W \{ F\theta + W^{-1}(y - \pi) \}.$$ 

Define new response vector $y' = F\theta + W^{-1}(y - \pi)$. The updated $\theta$ can be computed as a weighted least squares (WLS) estimates through model

$$y' = F\theta + \varepsilon, \quad \text{with } \varepsilon \sim (0, W^{-1}).$$

The associated WLS criterion is $\| W^{1/2} (y' - F\theta) \|^2$. Consequently, it can be immediately recognized in the spirit of Gauss-Newton approach that the updated $\theta$ can also be solved through a separable (weighted) nonlinear least squares problem $\| W^{1/2} (y' - S\beta) \|^2$ that is associated with model $y' = S\beta + \varepsilon$. This implies that the updated $\theta$ can also be solved via Algorithm 1, with response $y'$ and additional weights $W$. We may call this method as iteratively re-weighted separable nonlinear least squares (IRW-SNLS), as presented in Algorithm 2.
Algorithm 2 Iteratively Re-Weighted Separable Nonlinear Least Squares (IRWSNLS)

initialize $\theta = (\beta, c)^T$.

repeat
   compute quantities $s_i = s(c; x_i)$ and $\pi_i = \expit(\beta_0 + \beta_1 s_i)$ for $i = 1, \ldots, n$;
   form $\pi = [\pi_1, \ldots, \pi_n]$, $W = \{\pi_i(1 - \pi_i)\}$, $v = [-a \beta_1 s_i(1 - s_i)]$, and $F = \{j, s, v\}$;
   form new response vector $y' = F\theta + W^{-1}(y - \pi)$;
   update $\theta = \arg\min_{\theta} \| W^{1/2}(y' - S\beta) \|^2$ via Algorithm 1;
until convergence.

7 Proofs

7.1 Proof of Proposition 3.1

Omitted.

7.2 Proof of Proposition 4.1

As a result of the deflation step, we have $C(X) \perp C(r_\delta)$ and hence $C(X, \delta) = C(X) \oplus C(r_\delta)$. Since $P_{C(X, \delta)}y \in C(X, \delta)$, it follows that

$$P_{C(X, \delta)}y = P_{C(X)}y + P_{C(r_\delta)}y.$$ 

Now $Q(c)$ becomes

$$Q(c) = \| (y - P_{C(X)}y) - P_{C(r_\delta)}y \|^2 = \| r_y \|^2 - \| P_{C(r_\delta)}y \|^2 \equiv -\| P_{C(r_\delta)} \|^2$$

after simplification. Again, we have used notation $\equiv$ to denote equality up to some constant. For the last step, the first term is the residual sum of squares from Model (12), which does not involve the cutoff point $c$. Thus the best cutoff point

$$\hat{c} = \arg\min_c Q(c) = \arg\max_c \| P_{C(r_\delta)}y \|^2 .$$

More specifically, consider

$$\| P_{C(r_\delta)}y \|^2 = y^T P_{C(r_\delta)}y = y^T r_\delta (r_\delta^T r_\delta)^{-1} r_\delta^T y = y^T P_{C^\perp(X)}(\delta^T P_{C^\perp(X)}\delta)^{-1} \delta^T P_{C^\perp(X)}y$$

$$= \frac{(r_y^T \delta)^2}{\delta^T P_{C^\perp(X)} \delta} .$$

The last step is due to $P_{C^\perp(X)}y = r_y$.

For residual-based approach (Miller, 1996), the best cutoff point is sought by regressing residuals $r_y$ from Model (12) on $\delta$. The associated LS criterion is $\| P_{C^\perp(J, \delta)}r_y \|^2$. As previous, decompose
\( C(j, \delta) = C(j) \oplus C \left( \mathbf{P}_{C^\perp(j)} \delta \right) \). Thus
\[
\| \mathbf{P}_{C^\perp(j)} \mathbf{r}_y \|^2 = \| \mathbf{r}_y - \mathbf{P}_{C(j)} \mathbf{r}_y - \mathbf{P}_{C} \left( \mathbf{P}_{C^\perp(j)} \delta \right) \mathbf{r}_y \|^2
\]
\[
= - \| \mathbf{P}_{C} \left( \mathbf{P}_{C^\perp(j)} \delta \right) \mathbf{r}_y \|^2
\]
\[
= - \frac{(\mathbf{r}_y^T \delta)^2}{\delta^T \mathbf{P}_{C^\perp(j)} \delta}.
\]

The last step follows from \( \mathbf{P}_{C^\perp(j)} \mathbf{r}_y = \mathbf{r}_y \) since \( \mathbf{r}_y \in C^\perp(X) \subseteq C^\perp(j) \). The proof is completed. \( \blacksquare \)

References


