Time-to-event outcomes with BART

Rodney Sparapani
Medical College of Wisconsin

Robert McCulloch
Arizona State University

Abstract

This short article illustrates how to analyze time-to-event outcomes with the **BART** R package.

Keywords: Bayesian Additive Regression Trees.

1. Time-to-event outcomes with BART

The **BART** package supports time-to-event outcomes including survival analysis, competing risks and recurrent events.

1.1. Background: survival analysis with the Cox proportional hazard model

The inspiration for survival analysis with BART is the classic standard: the Cox proportional hazard model (Cox 1972). The data is $(s_i, \delta_i, \boldsymbol{x}_i)$ where s_i is the time of an absorbing event, $\delta_i = 1$, or right censoring, $\delta_i = 0$, and $\boldsymbol{x}_i$ is a vector of covariates (which can be time-dependent, but, for simplicity, we assume that they are known at time zero). We construct a grid of the ordered distinct event times, $0 = t_{(0)} < \cdots < t_{(J)} < \infty$, and we consider the following time intervals: $(0, t_{(1)}], (t_{(1)}, t_{(2)}], \ldots (t_{(J-1)}, t_{(J)}]$. The general form of the Cox proportional hazard model is the following: $\lambda(t_{(j)}, \boldsymbol{x}_i) = \lambda_0(t_{(j)}) \exp(\boldsymbol{\beta}'\boldsymbol{x}_i)$ where $\lambda(t_{(j)}, \boldsymbol{x}_i)$ is the hazard, $\lambda_0(t_{(j)})$ is a nonparametric baseline hazard defined at the grid of time points and $\exp(\boldsymbol{\beta}'\boldsymbol{x}_i)$ is a parametric multiplier which we call linear proportionality. To perform estimation and inference of $\boldsymbol{\beta}$, we utilize what is known as the partial likelihood: $[\boldsymbol{\beta}|\lambda_0(t)] = \prod_i \frac{e^{\boldsymbol{\beta}'\boldsymbol{x}_i}}{\sum_{j \in R(t_i)} e^{\boldsymbol{\beta}'\boldsymbol{x}_j}}$ where $R(t_i)$ is the set of subjects at risk for an event at time point t_i (which for events is a grid point by definition). The cumulative baseline hazard can be estimated as $\hat{\Lambda}_0(t_{(j)}) = \sum_{t_i \leq t_{(j)}} \frac{\delta_i}{\sum_{j \in R(t_i)} e^{\hat{\boldsymbol{\beta}}'\boldsymbol{x}_i}}$. The baseline survival is $\hat{S}_0(t_{(j)}) = e^{-\hat{\Lambda}_0(t_{(j)})}$ and the general survival is $\hat{S}(t_{(j)}, \boldsymbol{x}_i) = \hat{S}_0(t_{(j)})^{\exp(\hat{\boldsymbol{\beta}}'\boldsymbol{x}_i)}$. Notice that we don't directly estimate the survival; rather, we estimate $\boldsymbol{\beta}$ and survival is a consequence of this estimate by construction. This feature, and the time grid, foreshadow elements of survival analysis with BART.

1.2. Survival analysis with BART

Survival analysis with BART is provided by the `surv.bart` function for serial computation and `mc.surv.bart` for parallel computation. The complete details of our approach can be found in Sparapani, Logan, McCulloch, and Laud (2016) and a brief introduction follows. We

take an approach that is akin to discrete-time survival analysis (Fahrmeir 1998). Following the capabilities of BART, we do not stipulate linearity nor proportional hazards. We use the same notation developed above for the Cox proportional hazards model above.

Now, consider event indicators y_{ij} for each subject i at each distinct time $t_{(j)}$ up to and including the subject's observation time $t_i = t_{(n_i)}$ with $n_i = \#\{j : t_{(j)} \leq t_i\}$ or $n_i = \arg\max_j \left[t_{(j)} \leq t_i\right]$. This means $y_{ij} = 0$ if $j < n_i$ and $y_{in_i} = \delta_i$. Denote the probability of an event at time $t_{(j)}$, conditional on no previous event, by p_{ij}. Now, our model for y_{ij} is a nonparametric probit regression of y_{ij} on the time $t_{(j)}$ and the covariates $\boldsymbol{x}_i$. We utilize the Albert and Chib (1993) truncated Normal latent variables z_{ij} to recast it as a continuous BART model where the latents are the outcome. We choose Albert and Chib (1993) Normal latents as the default for computational efficiency, but we also provide the optional Holmes and Held (2006) Logistic latents by specifying `type='lbart'`.

So the model is

$$
\begin{aligned}
y_{ij} &= \delta_i \mathrm{I}\left(s_i = t_{(j)}\right), \ j = 1, \ldots, n_i \\
y_{ij}|p_{ij} &\sim \mathrm{B}\left(p_{ij}\right) \\
p_{ij}|f &= \Phi(\mu_{ij}), \ \mu_{ij} = \mu_0 + f(t_{(j)}, \boldsymbol{x}_i) \\
f &\overset{\text{prior}}{\sim} \text{BART} \\
z_{ij}|y_{ij}, f &\sim \begin{cases} \mathrm{N}\left(\mu_{ij}, \ 1\right) \mathrm{I}\left(-\infty, 0\right) & \text{if } y_{ij} = 0 \\ \mathrm{N}\left(\mu_{ij}, \ 1\right) \mathrm{I}\left(0, \infty\right) & \text{if } y_{ij} = 1 \end{cases}
\end{aligned}
$$

where Φ is the standard Normal cumulative distribution function.

If the event indicators, y_{ij}, are known, then you can specify them with the `y.train` argument (and, consequently, $\mu_0 = 0$, which is the default that you can over-ride with the `binaryOffset` argument). However, in most cases, these indicators would need to be constructed, so for convenience, you can specify (s_i, δ_i), with the arguments `times` and `delta` respectively. In this case, the default value of μ_0 assumes that the times follow the Exponential distribution and the covariates are not involved, i.e., $\mu_0 = \Phi^{-1}\left[1 - \exp\left(\frac{\sum_i \delta_i}{\sum_i s_i}\right)\right]$ (which you can over-ride with the `binaryOffset` argument). For BART with continuous outcomes, typically the outcome is centered and μ_0 is taken to be $\bar{y}$. While centering can be helpful for small samples with Albert and Chib (1993), it is unnecessary for moderate to large samples because of the flexibility of f (for Holmes and Held (2006) with Logistic latents which have heavier tails, centering is unnecessary even for small samples so μ_0 is fixed at zero if `type='lbart'`).

So just like in the Cox model case, we have to construct quantities of interest with BART for survival analysis. In discrete-time survival analysis, the probability of an event in an interval essentially replaces the instantaneous hazard in continuous-time survival analysis: $p(t, \boldsymbol{x}) = \Phi(\mu_0 + f(t, \boldsymbol{x}))$. And, the survival function is constructed as follows: $S(t_{(j)}|\boldsymbol{x}) = Pr(T > t_{(j)}|\boldsymbol{x}) = \prod_{l=1}^{j}(1 - p(t_{(l)}, \boldsymbol{x}))$.

Survival data pairs (s, δ) are converted to indicators by the helper function `surv.pre.bart` which is called automatically by `surv.bart` if `y.train` is not provided. `surv.pre.bart` returns a list which contains `y.train` for the indicators; `tx.train` for the covariates corresponding to `y.train` for training $f(t, \boldsymbol{x})$ (which includes time in the first column, and the rest of the covariates afterward, if any, i.e., rows of $[t, \boldsymbol{x}]$, hence the name `tx.train` to distinguish it from the original `x.train`); `tx.test` for the covariates to predict $f(t, \boldsymbol{x})$ rather

than to train; `times` which is the grid of ordered distinct time points; `K` which is the length of `times`; and `binaryOffset` which is μ_0. Here is a very simple example of a data set with three observations and no covariates re-formatted for display (no covariates is an interesting special case but we will discuss the more common case with covariates further below).

```
times <- c(2.5, 1.5, 3.0)
delta <- c(  1,   1,   0)
surv.pre.bart(times=times, delta=delta)
```

```
$y.train  $tx.train  $tx.test  $times    $K     $binaryOffset
[1]              t         t  [1]       [1] 3  [1] -0.6791459
    0     [1,] 1.5   [1,] 1.5    1.5
    1     [2,] 2.5   [2,] 2.5    2.5
    1     [3,] 1.5   [3,] 3.0    3.0
    0     [4,] 1.5
    0     [5,] 2.5
    0     [6,] 3.0
```

Here is a schematic of the input and output for the `surv.pre.bart` function.
`pre <- surv.pre.bart(times, delta, x.train, x.test=x.train)`

`pre` is a list with the matrix `pre$tx.train` & `pre$y.train` which is a vector

$$\begin{bmatrix} t_{(1)} & \boldsymbol{x}_1 \\ \vdots & \vdots \\ t_{(n_1)} & \boldsymbol{x}_1 \\ \vdots & \vdots \\ t_{(1)} & \boldsymbol{x}_N \\ \vdots & \vdots \\ t_{(n_N)} & \boldsymbol{x}_N \end{bmatrix} \begin{bmatrix} y_{11}=0 \\ \vdots \\ y_{1n_1}=\delta_1 \\ \vdots \\ y_{N1}=0 \\ \vdots \\ y_{Nn_N}=\delta_N \end{bmatrix}$$

For `pre$tx.test`, n_i is replaced by K which is very helpful so that each subject contributes an equal number of settings for programmatic convenience and noninformative estimation, i.e., if high-risk subjects with earlier events did not appear beyond their event, then estimates of survival for latter times would be biased upward. For other outcomes besides time-to-event, we provide two matrices of covariates, `x.train` and `x.test`, where `x.train` is for training and `x.test` is for validation. However, due to the variable n_i for time-to-event outcomes, we generally provide two arguments as follows: `x.train, x.test=x.train` where the former matrix will be expanded by `surv.pre.bart` to $\sum_{i=1}^{N} n_i$ rows for training $f(t, \boldsymbol{x})$ while the latter matrix will be expanded to $N \times K$ rows for $f(t, \boldsymbol{x})$ estimation only. If you still need to perform validation, then you can make a separate call to the `predict` function.

N.B. the argument `ndpost=M` is the length of the chain to be returned and the argument `keepevery` is used for thinning, i.e., return `M` observations where `keepevery` are culled in between each returned value. For BART with time-to-event outcomes, the default is `keepevery=10` (rather than `keepevery=1` for other outcomes) since the grid of time points inflates the size

of data sets and has a tendency towards higher auto-correlation, therefore, making thinning more necessary. To avoid unnecessarily massive data sets, coarsen the time axis appropriately.

Here is a schematic of the input and output for the `surv.bart` function for serial computation and `mc.surv.bart` for parallel computation.

```
set.seed(99)
post=surv.bart(x.train, times=times, delta=delta, x.test=x.train, ndpost=M) or
post=mc.surv.bart(x.train, times=times, delta=delta, x.test=x.train, ndpost=M,
mc.cores=C, seed=99)
```

Input vector `times` with `K` distinct values and `x.train`: $\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix}$ or $\boldsymbol{x}_i$

Output `post` of type `survbart` which is essentially a list of objects including the matrix: `post$surv.test`: $\hat{S}_m(t_{(j)}, \boldsymbol{x}_i)$

$$\begin{bmatrix} \hat{S}_1(t_{(1)},\boldsymbol{x}_1) & \ldots & \hat{S}_1(t_{(K)},\boldsymbol{x}_1) & \ldots & \hat{S}_1(t_{(1)},\boldsymbol{x}_N) & \ldots & \hat{S}_1(t_{(K)},\boldsymbol{x}_N) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{S}_M(t_{(1)},\boldsymbol{x}_1) & \ldots & \hat{S}_M(t_{(K)},\boldsymbol{x}_1) & \ldots & \hat{S}_M(t_{(1)},\boldsymbol{x}_N) & \ldots & \hat{S}_M(t_{(K)},\boldsymbol{x}_N) \end{bmatrix}$$

Here is a schematic of the input and output for the `predict.survbart` function.

```
pred <- predict(post, pre$tx.test, mc.cores=C)
```

Input: `x.test` $\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_Q \end{bmatrix}$ or $\boldsymbol{x}_i$

Output: `pred` of type `survbart` with `pred$surv.test`: $\hat{S}_m(t_{(j)}, \boldsymbol{x}_i)$

$$\begin{bmatrix} \hat{S}_1(t_{(1)},\boldsymbol{x}_1) & \ldots & \hat{S}_1(t_{(K)},\boldsymbol{x}_1) & \ldots & \hat{S}_1(t_{(1)},\boldsymbol{x}_Q) & \ldots & \hat{S}_1(t_{(K)},\boldsymbol{x}_Q) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{S}_M(t_{(1)},\boldsymbol{x}_1) & \ldots & \hat{S}_M(t_{(K)},\boldsymbol{x}_1) & \ldots & \hat{S}_M(t_{(1)},\boldsymbol{x}_Q) & \ldots & \hat{S}_M(t_{(K)},\boldsymbol{x}_Q) \end{bmatrix}$$

As previously noted, BART does not directly provide a summary of the effect of a single covariate, or a subset of covariates, on the outcome. For survival analysis, we use Friedman's partial dependence function (Friedman 2001) with BART to summarize the marginal effect due to a subset of the covariates, $(t, \boldsymbol{x}_S)$, by aggregating over the complement of covariates, $\boldsymbol{x}_C$, i.e., $\boldsymbol{x} = [\boldsymbol{x}_S, \boldsymbol{x}_C]$. The marginal dependence function is defined by fixing $(t, \boldsymbol{x}_S)$ while aggregating over the observed settings of the complement covariates in the cohort: $f(t, \boldsymbol{x}_S) = N^{-1} \sum_{i=1}^{N} f(t, \boldsymbol{x}_S, \boldsymbol{x}_{iC})$. For survival analysis, the f function is not directly of interest; rather, the survival function is more interpretable: $S(t, \boldsymbol{x}_S) = N^{-1} \sum_{i=1}^{N} S(t, \boldsymbol{x}_S, \boldsymbol{x}_{iC})$. Other marginal functions can be obtained in a similar fashion. Estimates can be derived via

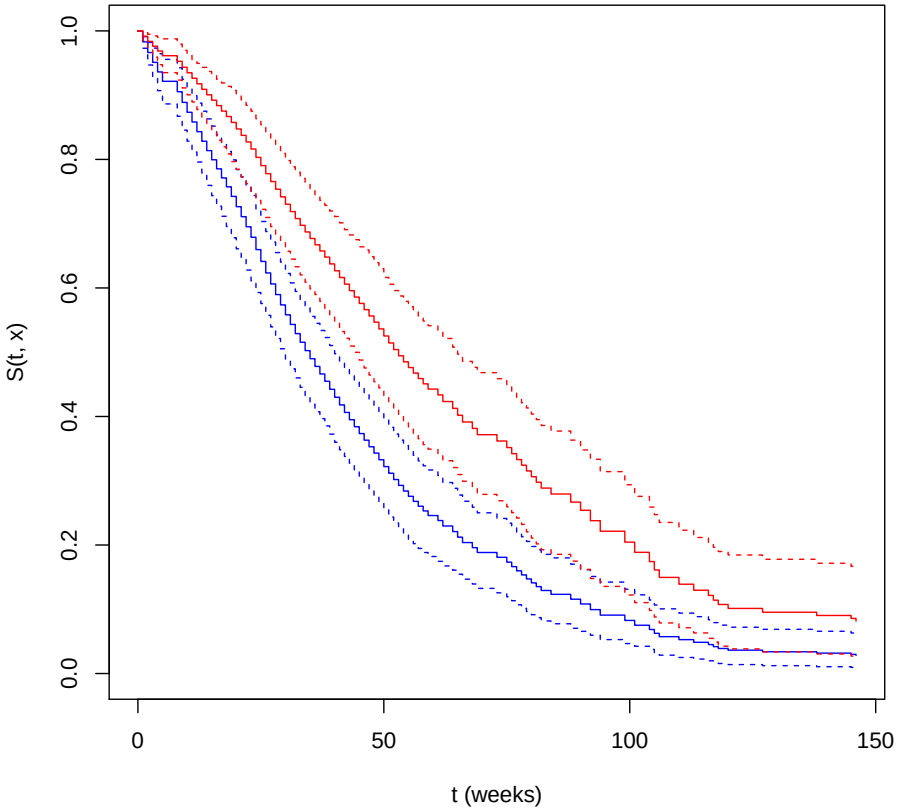

Figure 1: Advanced lung cancer example: Friedman's partial dependence function with 95% credible intervals: males (blue) vs. females (red).

functions of the posterior samples such as means, quantiles, e.g.,
$\hat{S}(t, \boldsymbol{x}_S) = M^{-1}N^{-1}\sum_{m=1}^{M}\sum_{i=1}^{N} S_m(t, \boldsymbol{x}_S, \boldsymbol{x}_{iC})$ where m indexes posterior samples. Friedman's partial dependence function is a concept that is very flexible. So flexible that we are unable to provide abstract functional support in the **BART** package; rather, we provide examples of the many practical uses in the `demo` directory.

Here we present an example that is available in the **BART** package.
`system.file('demo/lung.surv.bart.R', package='BART')`. The North Central Cancer Treatment Group surveyed 228 advanced lung cancer patients (Loprinzi, Laurie, Wieand, Krook, Novotny, Kugler, Bartel, Law, Bateman, and Klatt 1994). This data can be found in the `lung` data set. The study focused on prognostic variables. Patient responses were paired with a few clinical variables. We control for age, gender and Karnofsky performance score as rated by their physician. We compare the survival for males and females with Friedman's partial dependence function; see Figure 1. We also analyze this data set with Logistic latents and the results are quite similar (not shown): `system.file('demo/lung.surv.lbart.R', package='BART')`. Furthermore, we perform convergence diagnostics on the chain:

`system.file('demo/geweke.lung.surv.bart.R', package='BART')`.

1.3. Competing risks with BART

Competing risks are supported by the function `crisk.bart` for serial computation and `mc.crisk.bart` for parallel computation. Typically, competing risks (Fine and Gray 1999; Kalbfleisch and Prentice 2002) deal with events which are mutually exclusive, say, death from cardiovascular disease vs. death from other causes, i.e., a patient experiencing one of the events is prevented from experiencing another. We adopt the subdistribution concept of competing risks from Fine and Gray (1999). Let's suppose we have two kinds of events: kind 1 events, death from cause 1 which is the cause of interest, and kind 2 events, death from cause 2 which is any other cause. The distribution function of an event time is $F(t, \boldsymbol{x}) = G_1(t, \boldsymbol{x}) + G_2(t, \boldsymbol{x})$ where $G_1(t, \boldsymbol{x}) = pF_1(t, \boldsymbol{x})$ and $G_2(t, \boldsymbol{x}) = (1-p)F_2(t, \boldsymbol{x})$. F_1 and F_2 are distribution functions which integrate to one and G_1 and G_2 are subdistribution functions which do not. Fine and Gray (1999) model the subdistribution functions rather than the distribution functions; and we will do the same. But, here we part ways with Fine and Gray (1999) since they assume linear proportionality and the Exponential distribution while we impose neither precarious restrictive assumption.

To accomodate competing risks, we adapt our notation slightly: (s_i, δ_i) where $\delta_i = 1$ for kind 1 events, $\delta_i = 2$ for kind 2 events, or $\delta_i = 0$ for censoring times. We create a single grid of time points for the ordered distinct times based on either kind of event or censoring: $0 = t_{(0)} < t_{(1)} < \cdots < t_{(J)} < \infty$. We model the probability of a kind 1 event, $p_1(t_{(j)}, \boldsymbol{x}_i)$, and kind 2 events conditioned on subject i being alive at time $t_{(j)}$, $p_2(t_{(j)}, \boldsymbol{x}_i)$. Now, as before, we create event indicators.

$$
\begin{aligned}
y_{1ij} &= \mathrm{I}(\delta_i = 1)\,\mathrm{I}(j = n_i) \text{ where } j = 1, \ldots, n_i \\
y_{1ij} | p_{1ij} &\sim \mathrm{B}(p_{1ij}) \\
p_{1ij} &= \Phi(\mu_1 + f_1(t_{(j)}, \boldsymbol{x}_i) \text{ where } f_1 \overset{\text{prior}}{\sim} \text{BART} \\
y_{2ij} &= \mathrm{I}(\delta_i = 2)\,\mathrm{I}(j = n_i) \text{ where } j = 1, \ldots, n_i - y_{1in_i} \\
y_{2ij} | p_{2ij} &\sim \mathrm{B}(p_{2ij}) \\
p_{2ij} &= \Phi(\mu_2 + f_2(t_{(j)}, \boldsymbol{x}_i) \text{ where } f_2 \overset{\text{prior}}{\sim} \text{BART}
\end{aligned}
$$

Based on this BART framework, we can estimate the survival function and the cumulative incidence functions as follows.

$$
S(t, \boldsymbol{x}_i) = 1 - F(t, \boldsymbol{x}_i) = \prod_{j=1}^{k} (1 - p_{1ij})(1 - p_{2ij}) \text{ where } k = \arg\max_j \left[t_{(j)} \leq t \right]
$$

$$
F_1(t, \boldsymbol{x}_i) = \int_0^t S(u-, \boldsymbol{x}_i) \lambda_1(u, \boldsymbol{x}_i) \mathrm{d}u = \sum_{j=1}^{k} S(t_{(j-1)}, \boldsymbol{x}_i) p_{1ij}
$$

$$
F_2(t, \boldsymbol{x}_i) = \int_0^t S(u-, \boldsymbol{x}_i) \lambda_2(u, \boldsymbol{x}_i) \mathrm{d}u = \sum_{j=1}^{k} S(t_{(j-1)}, \boldsymbol{x}_i)(1 - p_{1ij}) p_{2ij}
$$

The returned object of type `criskbart` from `crisk.bart` or `mc.crisk.bart` provides the cumulative incidence functions and survival corresponding to `x.test` as follows: F_1 is `cif.test`, F_2 is `cif.test2` and S is `surv.test`.

Here, we present the Mayo Clinic liver transplant waiting list data from 1990-1999 with $N = 815$ patients. During the study period, the liver transplant organ allocation policy was flawed. Blood type is an important matching factor to avoid organ rejection. Donor livers from subjects with blood type O can be used by patients with A, B, AB or O blood types; whereas a donor liver from the other types will only be transplanted to a matching A, B or AB recipient. Therefore, type O subjects on the waiting list were at a disadvantage since the pool of competitors was larger for type O donor livers. This data is of historical interest and provides a useful example of competing risks, but it has little relevance today. Current liver transplant policies have evolved and now depend on each individual patient's risk/need which are assessed and updated regularly while a patient is on the waiting list. However, there still remains an acute shortage of donor livers today. The `transplant` data set is provided by the **BART** R package as is this example: `system.file('demo/liver.crisk.bart.R', package='BART')`. We compare the nonparametric Aalen-Johansen competing risks estimator with BART for the transplant event of type O patients which are in general agreement; see Figure 2.

1.4. Recurrent events with BART

The **BART** package supports recurrent events with `recur.bart` for serial computation and `mc.recur.bart` for parallel computation. Due to the capabilities of BART, we have great flexibility in modeling the dependence of recurrent events on covariates. Consider data in the form: $\delta_i, s_i, \boldsymbol{t}_i, \boldsymbol{u}_i, \boldsymbol{x}_i(t)$ where $i = 1, \ldots, n$ indexes subjects; s_i is the end of the observation period (death, $\delta_i = 1$, or censoring, $\delta_i = 0$); N_i is the number of events during the observation period; $\boldsymbol{t}_i = [t_{i1}, \ldots, t_{iN_i}]$ and t_{ik} is the event start time of the kth event (let $t_{i0} = 0$); $\boldsymbol{u}_i = [u_{i1}, \ldots, u_{iN_i}]$ and u_{ik} is the event end time of the kth event (let $u_{i0} = 0$); and $\boldsymbol{x}_i(t)$ is a vector of time-dependent covariates. Both start and end times of events are necessary to define risk set eligibility for non-absorbing events like readmissions since patients currently hospitalized cannot be readmitted. For instantaneous events (or roughly instantaneous events such as emergency department visits with time measured in days), the end times are ignored.

We denote the J collectively distinct event start and end times for all subjects by $0 < t_{(1)} < \cdots < t_{(J)} < \infty$ thus taking $t_{(j)}$ to be the j^{th} order statistic among distinct observation times and, for convenience, $t_{(j')} = 0$ where $j' \leq 0$ (note that $t_{(j)}$ are constructed from all event start/end times for all subjects, but they may be a censoring time for any given subject). Now consider binary event indicators y_{ij} for each subject i at each distinct time $t_{(j)}$ up to the subject's last observation time $t_{(n_i)} \leq s_i$ with $n_i = \arg\max_j \left[t_{(j)} \leq s_i\right]$, i.e., $y_{i1}, \ldots, y_{in_i} \in \{0, 1\}$. We then denote by p_{ij} the probability of an event at time $t_{(j)}$ conditional on $\left(t_{(j)}, \tilde{\boldsymbol{x}}_i(t_{(j)})\right)$ where $\tilde{\boldsymbol{x}}_i(t_{(j)}) = \left(N_i(t_{(j-1)}), v_i(t_{(j)}), \boldsymbol{x}_i(t_{(j)})\right)$. Let $N_i(t-) \equiv \lim_{s \uparrow t} N_i(s)$ be the counting process of events for subject i just prior to time t and we also define $N_i = N_i(s_i)$. Let $v_i(t) = t - u_{N_i(t-)}$ is the sojourn time for subject i, i.e., time since last event, if any. Notice that we can replace $N_i(t_{(j)}-)$ with $N_i(t_{(j-1)})$ since, by design, the state of information available at time $t_{(j)}-$ is the same as that available at $t_{(j-1)}$. Assuming a constant intensity and constant covariates, $\tilde{\boldsymbol{x}}_i(t_{(j)})$, in the interval $(t_{(j-1)}, t_{(j)}]$, we define the cumulative intensity

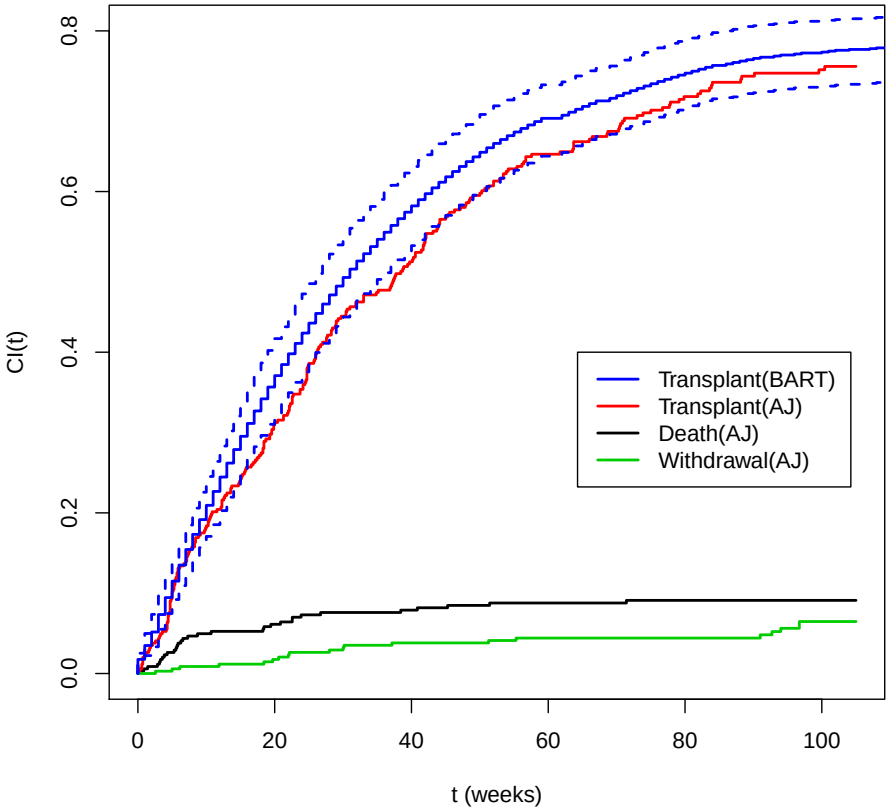

Figure 2: Liver transplant competing risks for type O patients estimated by BART and Aalen-Johansen.

process as:

$$\Lambda(t_{(j)}, \tilde{\boldsymbol{x}}_i(t_{(j)})) = \int_0^{t_{(j)}} \mathrm{d}\Lambda(t, \tilde{\boldsymbol{x}}_i(t)) = \sum_{j'=1}^{j} \Pr N_i(t_{(j')}) - N_i(t_{(j'-1)}) = 1 \mid t_{(j')}, \tilde{\boldsymbol{x}}_i(t_{(j')}) = \sum_{j'=1}^{j} p_{ij'} \tag{1}$$

where, for the moment, we leave these p_{ij} unspecified; we will provide a definition later (2). Note: the terms "intensity" and "hazard" are generally interchangeable.

With absorbing events, you need not worry about the conditional independence of future events (there aren't any); but, with recurrent events, we need to address it. Conditional independence can be had by conditioning on the entire event history, $N_i(s)$ where $0 \leq s < t$, which would be the most general model. However, conditioning on the entire event history is often problematic as it is here. We condition on both $N_i(t-)$ and the most recent sojourn time, $v_i(t)$, which gives us a reasonable expectation of conditional independence.

We now write the model for y_{ij} as a nonparametric probit regression of y_{ij} on $\left(t_{(j)}, \tilde{\boldsymbol{x}}_i(t_{(j)})\right)$ akin to parametric models of discrete-time intensity (Fahrmeir 1998). Specifically, with temporal data converted from $\delta_i, s_i, \boldsymbol{t}_i, \boldsymbol{u}_i, \boldsymbol{x}_i(t)$ to a sequence of longitudinal binary events as follows: $y_{ij} = \max_k \mathrm{I}\left(t_{ik} = t_{(j)}\right)$. But, what is the range of j in the equation above? As we shall see, there is a caveat in translating this into the likelihood which we address by a vignette.

Suppose that we have two subjects with the following values:

$$\begin{aligned} N_1 = 2, s_1 = 9, t_{11} = 3, u_{11} = 7, t_{12} = 8, u_{12} = 8 \Rightarrow y_{11} = 1, y_{12} = y_{13} = 0, y_{14} = 1, y_{15} = 0 \\ N_2 = 1, s_2 = 12, t_{21} = 4, u_{21} = 7 \Rightarrow y_{21} = 0, y_{22} = 1, y_{23} = y_{24} = y_{25} = y_{26} = 0 \end{aligned} \tag{2.3}$$

which creates the grid of times $(3, 4, 7, 8, 9, 12)$. For subject 1 (2), notice that $y_{12} = y_{13} = 0$ ($y_{23} = 0$) as it should be since no event occurred at times 4 or 7 (7). However, no events occurred because their first event had not ended yet, i.e., these time points do not contribute to the likelihood since these subjects are not chronologically at risk for an event. The **BART** package provides the `recur.pre.bart` function which you can use to construct these data sets. Here is a short demonstration of its capabilities adapted from `demo/data.recur.pre.bart.R` (re-formatted for display purposes).

```
> library(BART)
> times <- matrix(c(3, 8, 9, 4, 12, 12), nrow=2, ncol=3, byrow=TRUE)
> tstop <- matrix(c(7, 8, 0, 7,  0,  0), nrow=2, ncol=3, byrow=TRUE)
> delta <- matrix(c(1, 1, 0, 1,  0,  0), nrow=2, ncol=3, byrow=TRUE)
> recur.pre.bart(times=times, delta=delta, tstop=tstop)
$y.train                $tx.train     $tx.test
[1] 1 1 0 0 1 0 0 0           t v N         t v N
$times                  [1,]  3 3 0   [1,]  3 3 0
[1]  3  4  7  8  9 12   [2,]  8 5 1   [2,]  4 1 1
$K                      [3,]  9 1 2   [3,]  7 4 1
[1] 6                   [4,]  3 3 0   [4,]  8 5 1
                        [5,]  4 4 0   [5,]  9 1 2
                        [6,]  8 4 1   [6,] 12 4 2
```

```
[7,]  9 5 1   [7,]  3 3 0
[8,] 12 8 1   [8,]  4 4 0
              [9,]  7 3 1
             [10,]  8 4 1
             [11,]  9 5 1
             [12,] 12 8 1
```

Notice that `$tx.test` is not limited to the same time points as `$tx.train`, i.e., we often want/need to estimate f at counter-factual values not observed in the data so each subject contributes an equal number of evaluations for estimation purposes.

Therefore, the only y_{ij} that contribute to the likelihood are those corresponding to $j \in R_i(t_{(j)})$ where the risk set for subject i at time $t_{(j)}$ either contains j or is empty. We define the risk set as $R_i(t_{(j)}) = \left\{j : (j \in \{1, \ldots, n_i\}) \cap \left(\cap_{k=1}^{N_i}\{t_{(j)} \notin (t_{ik}, u_{ik})\}\right)\right\}$ and we denote the binary outcomes for subject i by the longitudinally ordered vector $\boldsymbol{y}_i = [y_{ij}]$ where $j \in R_i(t_{(j)})$.

Putting it all together, we arrive at the following recurrent events discrete-time model.

$$\begin{aligned} y_{ij}|p_{ij} &\sim \mathrm{B}\left(p_{ij}\right) \text{ where } i = 1, \ldots, n; j \in R_i(t_{(j)}) \\ p_{ij}|f &= \Phi(\mu_{ij}),\ \mu_{ij} = \mu_0 + f(t_{(j)}, \tilde{\boldsymbol{x}}_i(t_{(j)})) \\ f &\overset{\text{prior}}{\sim} \text{BART} \end{aligned} \tag{2}$$

For computational efficiency, we carry out the probit regression via truncated Normal latent variables z_{ij} to reduce it to a continuous outcome BART model like so (Albert and Chib 1993) (this default can be over-ridden to utilize Holmes and Held (2006) Logistic latents by specifying `type='lbart'`).

$$z_{ij}|y_{ij}, f \sim \begin{cases} \mathrm{N}\left(\mu_{ij},\ 1\right)\mathrm{I}\left(-\infty, 0\right) & \text{if } y_{ij} = 0 \\ \mathrm{N}\left(\mu_{ij},\ 1\right)\mathrm{I}\left(0, \infty\right) & \text{if } y_{ij} = 1 \end{cases}$$

Consequently, we have the following likelihood where $\boldsymbol{y}$ (the entire collection of $\boldsymbol{y}_i$'s) is given $\boldsymbol{p}$ (the entire collection of p_{ij}'s): $[\boldsymbol{y}|\boldsymbol{p}] = \prod_{i=1}^{n}\prod_{j\in R_i} p_{ij}^{y_{ij}}(1-p_{ij})^{1-y_{ij}}$. For binary data, $\mu_0 = \Phi^{-1}(p_0)$ can be used for centering the latents around the probability of an event p_0. For recurrent event data, we can similarly center the latents by assuming the times of recurrent events follow the Exponential distribution and the covariates, $\tilde{\boldsymbol{x}}$, have no impact, i.e., $\mu_0 = \Phi^{-1}\left(1 - \exp\left(\frac{\sum_i N_i}{\sum_i s_i}\right)\right)$.

With the data prepared as described in the above vignette, the BART model for binary data treats the probability of an event within an interval as a nonparametric function of time, t, and covariates, $\tilde{\boldsymbol{x}}(t)$. Conditioned on the data, BART provides samples from the posterior distribution of f. For any t and $\tilde{\boldsymbol{x}}(t)$, we obtain the posterior distribution of $p(t, \tilde{\boldsymbol{x}}(t)) = \Phi(\mu_0 + f(t, \tilde{\boldsymbol{x}}(t)))$.

For the purposes of recurrent events survival analysis, we are typically interested in estimating the cumulative intensity function as presented in formula (1). With these estimates, one can accomplish inference from the posterior via means, quantiles or other functions of $p(t, \tilde{\boldsymbol{x}}_i(t))$ or $\Lambda(t, \tilde{\boldsymbol{x}}(t))$ as needed such as the relative intensity, i.e., $RI(t, \tilde{\boldsymbol{x}}_n(t), \tilde{\boldsymbol{x}}_d(t)) = \frac{p(t,\tilde{\boldsymbol{x}}_n(t))}{p(t,\tilde{\boldsymbol{x}}_d(t))}$ where $\tilde{\boldsymbol{x}}_n(t)$ and $\tilde{\boldsymbol{x}}_d(t)$ are two settings we wish to compare like two treatments.

An interesting example of recurrent events involves a clinical trial conducted by the Veterans Administration Cooperative Urological Research Group. In this study, all patients had superficial bladder tumors when they entered the trial. These tumors were removed transurethrally and patients were randomly assigned to one of three treatments: placebo, thiotepa or pyridoxine (vitamin B6). Many patients had multiple recurrences of tumors during the study and new tumors were removed at each visit. For each patient, their recurrence time, if any, was measured from the beginning of treatment. There were 118 patients enrolled but only 116 were followed beyond time zero and contribute information. This data set is loaded by `data(bladder)` and the loaded data frame of interest is `bladder1`. This data set is analyzed by `system.file('demo/bladder.recur.bart.R', package='BART')`. In Figure 3, notice that the relative intensity calculated by Friedman's partial dependence function favors thiotepa over placebo from roughly 3 to 18 months and afterward they are about equal, but the 95% credible intervals are wide throughout. Similarly, the relative intensity calculated by Friedman's partial dependence function favors thiotepa over vitamin B6 from roughly 3 to 18 months and afterward they are about equal, but the 95% credible intervals are wide throughout; see Figure 4. And, finally, vitamin B6 is no better than placebo, and possibly worse, but the 95% credible intervals are wide; see Figure 5.

References

Albert J, Chib S (1993). "Bayesian analysis of binary and polychotomous response data." *JASA*, **88**, 669–79.

Cox DR (1972). "Regression models and life tables (with Discussion)." *Journal of the Royal Statistical Society, Series B*, **34**, 187–220.

Fahrmeir L (1998). "Discrete survival-time models." In *Encyclopedia of biostatistics*, pp. 1163–1168. Wiley, Chichester.

Fine JP, Gray RJ (1999). "A proportional hazards model for the subdistribution of a competing risk." *Journal of the American statistical association*, **94**, 496–509.

Friedman JH (2001). "Greedy function approximation: a gradient boosting machine." *Annals of Statistics*, **29**, 1189–1232.

Holmes C, Held L (2006). "Bayesian auxiliary variable models for binary and multinomial regression." *Bayesian Analysis*, **1**, 145–68.

Kalbfleisch J, Prentice R (2002). *The statistical analysis of failure time data*. second edition. Wiley.

Loprinzi CL, Laurie JA, Wieand HS, Krook JE, Novotny PJ, Kugler JW, Bartel J, Law M, Bateman M, Klatt NE (1994). "Prospective evaluation of prognostic variables from patient-completed questionnaires. North Central Cancer Treatment Group." *Journal of Clinical Oncology*, **12**(3), 601–607.

Sparapani RA, Logan BR, McCulloch RE, Laud PW (2016). "Nonparametric survival analysis using Bayesian Additive Regression Trees (BART)." *Statistics in medicine*, **35**(16), 2741–53.

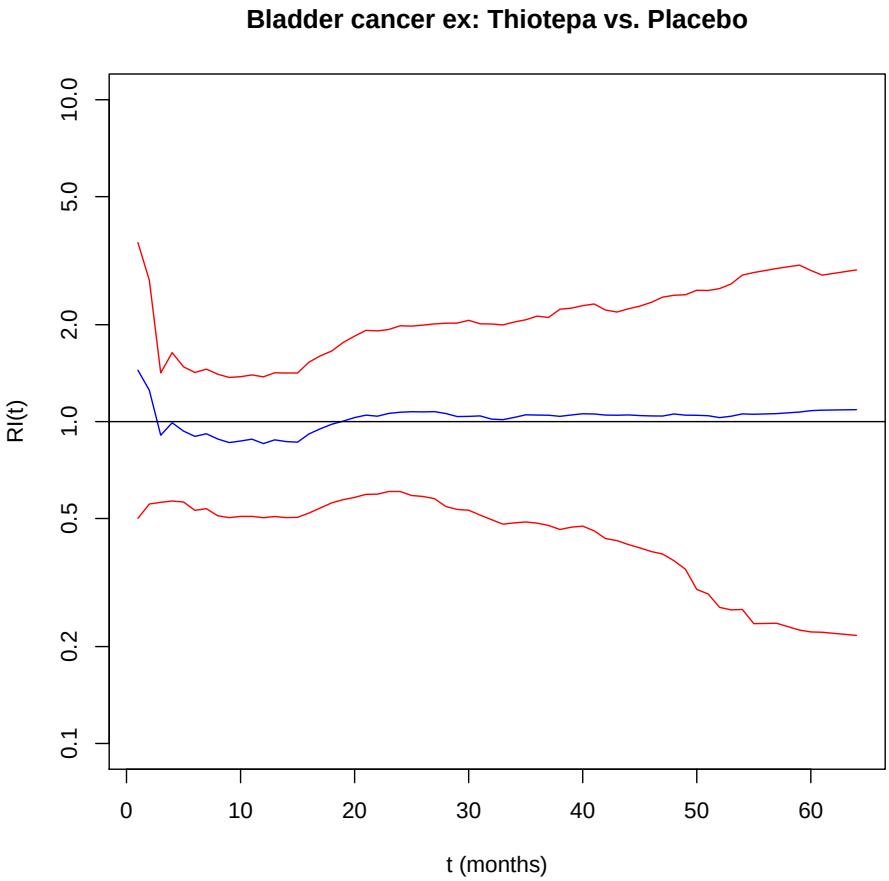

Figure 3: Relative Intensity via Friedman's partial dependence function: Thiotepa vs. Placebo.

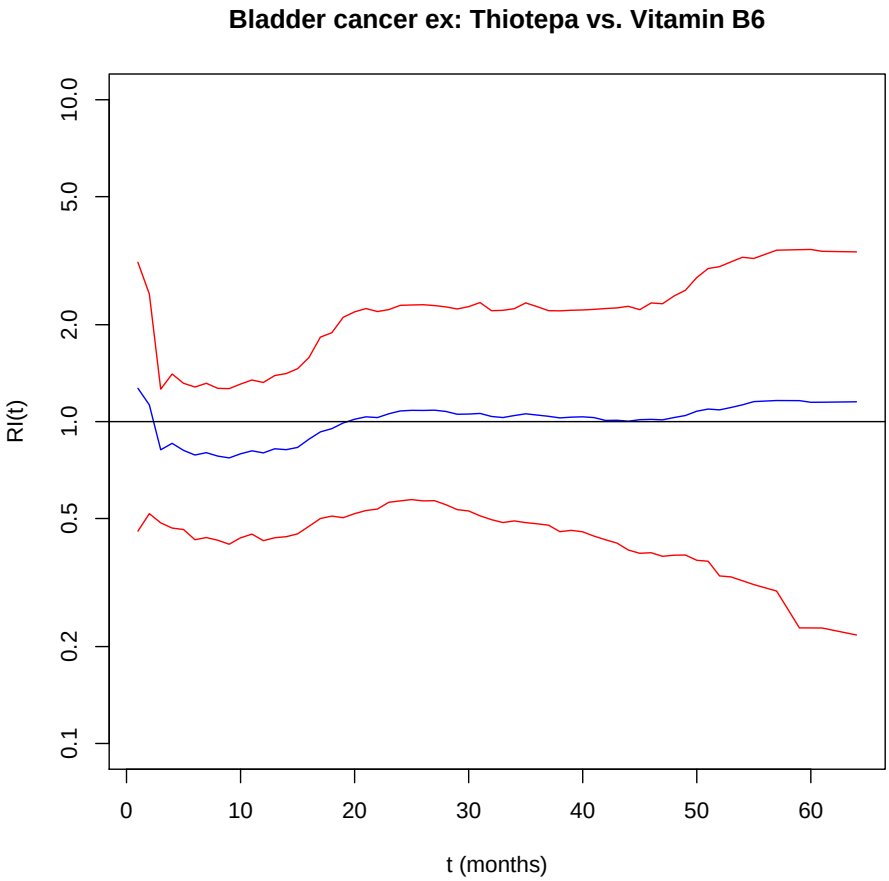

Figure 4: Relative Intensity via Friedman's partial dependence function: Thiotepa vs. Vitamin B6.

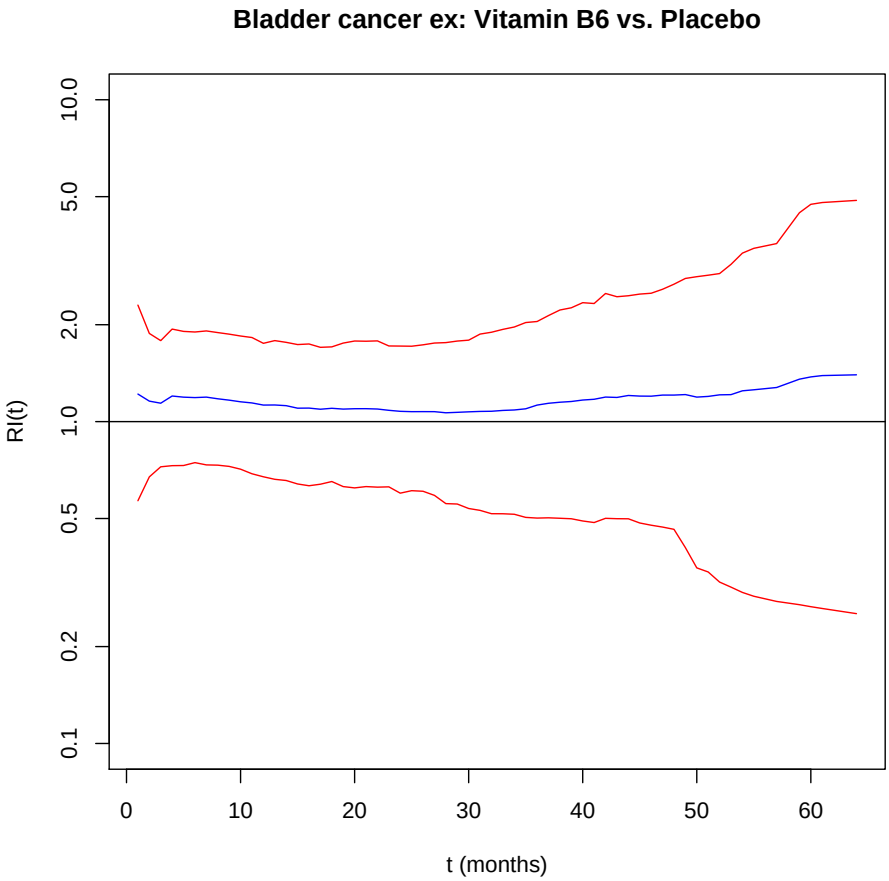

Figure 5: Relative Intensity via Friedman's partial dependence function: Vitamin B6 vs. Placebo.

Affiliation:

Rodney Sparapani rsparapa@mcw.edu
Division of Biostatistics, Institute for Health and Equity
Medical College of Wisconsin, Milwaukee campus