Biostatistics (2021) **22**, 1, *pp.* 198–213
doi:10.1093/biostatistics/kxz025
Advance Access publication on July 13, 2019

An ensemble method for interval-censored time-to-event data

WEICHI YAO*, HALINA FRYDMAN, JEFFREY S. SIMONOFF

Department of Technology, Operations, and Statistics, Stern School of Business, New York University, 44 West 4th Street, New York, NY, USA

wyao@stern.nyu.edu

Summary

Interval-censored data analysis is important in biomedical statistics for any type of time-to-event response where the time of response is not known exactly, but rather only known to occur between two assessment times. Many clinical trials and longitudinal studies generate interval-censored data; one common example occurs in medical studies that entail periodic follow-up. In this article, we propose a survival forest method for interval-censored data based on the conditional inference framework. We describe how this framework can be adapted to the situation of interval-censored data. We show that the tuning parameters have a non-negligible effect on the survival forest performance and guidance is provided on how to tune the parameters in a data-dependent way to improve the overall performance of the method. Using Monte Carlo simulations, we find that the proposed survival forest is at least as effective as a survival tree method when the underlying model has a tree structure, performs similarly to an interval-censored Cox proportional hazards model fit when the true relationship is linear, and outperforms the survival tree method and Cox model when the true relationship is nonlinear. We illustrate the application of the method on a tooth emergence data set.

Keywords: Conditional inference survival forest; Cox model; Data-dependent tuning parameters; Interval-censored data; Survival data; Survival tree method.

1. Introduction

Most statistical methods for the analysis of survival time (time-to-event) data have been developed in the situation where the observations could be right-censored. In many situations, however, the survival time cannot be directly observed and it is only known to have occurred in an interval obtained from a sequence of examination times. In this situation, we say that the survival time is interval-censored.

Interval-censored data are encountered in many medical and longitudinal studies, and various methods have been developed for their analysis. Finkelstein (1986) provided the first method for estimation of the Cox proportional hazard model from interval-censored data. Surveys of later approaches to the estimation of the Cox model and other semi- or parametric-survival models for interval-censored data can be found in Sun (2006) and Bogaerts *and others* (2017). However, these methods rely on restrictive assumptions

*To whom correspondence should be addressed.

such as proportional hazards and a log-linear relationship between the hazard function and covariates. Furthermore, because these methods are often parametric, nonlinear effects of variables must be modeled by transformations or expanding the design matrix to include specialized basis functions for more complex data structures in real-world applications.

Recently, Fu and Simonoff (2017) proposed a nonparametric recursive-partitioning (tree) method for interval-censored survival data, as an extension of the conditional inference tree method for right-censored data of Hothorn *and others* (2006b). As is well known, tree estimators are nonparametric and as such often exhibit low bias and high variance. Compared to simple models like trees, ensemble methods like bagging and random forest can reduce variance while preserving low bias. These methods average over predictions of the base learners (the trees) that have been fit to bootstrap samples, and are able to remain stable in high-dimensional settings and therefore can substantially improve prediction performance (Breiman, 2001). Ishwaran *and others* (2008) proposed the random survival forest (RSF) that extends random forest (Breiman, 2001) to right-censored survival data. Hothorn *and others* (2006a) proposed the conditional inference survival forest (with the conditional inference survival tree as the base learner) by incorporating weights into random forest-like algorithms and extending gradient boosting in order to minimize a weighted form of the empirical risk.

In this article, we propose a conditional inference survival forest method appropriate for interval-censored data (we will refer to this method as the IC cforest method). The goal of this ensemble tree algorithm is to lower the variance compared to an individual tree and therefore stabilize and improve the prediction performance. The proposed method is an extension of the conditional inference forest method (which is designed to handle right-censored survival data, and will be referred as the cforest method) with the base learner being the conditional inference survival tree proposed by Fu and Simonoff (2017) (we will refer to this as the IC ctree method).

2. An interval-censored survival forest

2.1. *Extending the survival forest of Hothorn and others (2006a)*

The recursive partitioning proposed in Hothorn *and others* (2006b) for building the ctree is based on a test of the global null hypothesis of independence between response variable $\mathbf{Y}$ and any of the m covariates $X_1, ..., X_m$. As a decision tree-based ensemble method, cforest induces randomness into each node of each individual tree (that is built from a bootstrap sample) when selecting a variable to split on. Only a random subset of covariates is considered for splitting at each node. The recursive partitioning in cforest is based on a test of the global null hypothesis of independence between response variable $\mathbf{Y}$ and any of the elements in a random subset I of the total m covariates (indeed, the size of this random subset $|I|$ is prespecified, with further discussion given in Section 2.2). In each node, after such a random subset I is selected, permutation-based multiple testing procedures are applied. The recursion stops if the global null hypothesis of independence cannot be rejected at a prespecified level α. If it can be rejected, the association between $\mathbf{Y}$ and each of the covariates $X_j, j \in I$ is measured to select the covariate with strongest association to the response variable $\mathbf{Y}$ (the one with minimum p-value, indicating the largest deviation from the partial null hypotheses). Once a covariate is selected, the permutation test framework is again used to find the optimal binary split.

The $|I|$-dimensional covariate vector $\mathbf{X}_I = (X_j)_{j \in I}$ falls in a space denoted by $\mathcal{X}_I = \prod_{j \in I} \mathcal{X}_j$, and $\mathbf{Y} \in \mathcal{Y}$. The association of the response variable $\mathbf{Y}$ and a predictor $X_j, j \in I$ based on a random sample $\mathcal{L}_n = \{(\mathbf{Y}_i, X_{1i}, X_{2i}, ..., X_{mi});\ i = 1, ..., n\}$ is measured by linear statistics of the form

$$T_j(\mathcal{L}_n, \mathbf{w}) = vec\left(\sum_{i=1}^{n} w_i g_j(X_{ji}) h(\mathbf{Y}_i, (\mathbf{Y}_1, ..., \mathbf{Y}_n))^T\right) \in \mathbb{R}^{p_j q},$$

where $\mathbf{w} := (w_1, ..., w_n)$ is a vector of non-negative integer-valued case weights having nonzero elements when the corresponding observations are elements of the node and zero otherwise, $g_j : \mathcal{X}_j \to \mathbb{R}^{p_j}$ is a nonrandom transformation of covariate X_j, and $h : \mathcal{Y} \times \mathcal{Y}^n \to \mathbb{R}^q$ is the influence function and depends on the responses $(\mathbf{Y}_1, ..., \mathbf{Y}_n)$ in a permutation-symmetric way. The dimension $p_j, j \in I$ and q vary according to different practical settings. Numeric covariates can be handled by the identity transformation $g_{ji}(x) = x$ with $p_j = 1$. Nominal covariates at levels $1, ..., K$ are represented by $g_{ji}(k) = e_K(k)$, the unit vector of length K with kth element being equal to one, and then $p_j = K$. For censored regression, the influence function h may be chosen as log-rank scores taking censoring into account, in which case $q = 1$. In their extension of ctree to IC ctree, Fu and Simonoff (2017) specified the influence function h to be the log-rank score for interval-censored data proposed by Pan (1998). This score assigns a univariate scalar value U_i to the bivariate response $\mathbf{Y}_i = (L_i, R_i]$, where L_i and R_i are the left and right endpoints of the censoring interval for the ith observation. It is defined as

$$U_i = \frac{\widehat{S}(L_i)\log\widehat{S}(L_i) - \widehat{S}(R_i)\log\widehat{S}(R_i)}{\widehat{S}(L_i) - \widehat{S}(R_i)}, \quad \text{when } L_i < R_i$$

and

$$U_i = 1 + \log\widehat{S}(L_i), \quad \text{when } L_i = R_i,$$

where $\widehat{S}(\cdot)$ is the nonparametric maximum likelihood estimator (NPMLE) of the survival function. We similarly use the log-rank score U_i in our proposed extension of cforest to IC cforest.

The aggregation scheme of the cforest is different from that of the RSF. Instead of averaging predictions directly as in the RSF, it works by averaging observation weights extracted from each of the individual trees and estimates the conditional survival probability function by computing one single estimator of the survival curve (the Kaplan–Meier curve for right-censored survival data and NPMLE for interval-censored data) based on weighted observations identified by the leaves of bootstrap survival trees. The idea of averaging weights instead of predictions is advocated in Meinshausen (2006) for quantile regression. Athey *and others* (2019) also adopt the same scheme for more general settings and propose the generalized random forest. These weights can be viewed as "adaptive nearest neighbor weights," a term borrowed from Lin and Jeon (2006), where these weights were theoretically studied for the estimation of conditional means for regression forests. The core idea is to obtain a "distance" or a "similarity" measure based on the number of times a pair of observations is assigned to the same terminal node in the different trees of the forest. For conditional mean estimation, the averaging and weighting views of forests are equivalent; however, if we move to more general settings like constructing a nonparametric method for complex data situations, the weighting scheme has been proved to be more efficient (Athey *and others*, 2019).

Consider cforest where a set of B trees is grown, indexed by $b = 1, 2, ..., B$. Each leaf of a tree corresponds to a rectangular subspace of $\mathcal{X}$. For any new observation $\mathbf{x} \in \mathcal{X}$, for each tree there is one and only one leaf such that $\mathbf{x}$ falls into it. Denote the corresponding rectangular subspace of this leaf in the bth tree as $R_b(\mathbf{x}) \subseteq \mathcal{X}$. The weight of each observation $\mathbf{X}_i = (X_{1i}, ..., X_{mi})^T$ in the original sample, $v_{i,b}(\mathbf{x})$, measures the "similarity" of the ith observation $\mathbf{X}_i$ to the new observed value $\mathbf{x}$ by counting how many times the value of $\mathbf{X}_i$ in the original sample falls into the same leaf as $\mathbf{x}$ in the bth tree

$$v_{i,b}(\mathbf{x}) = \frac{\mathbf{1}_{\{\mathbf{X}_i \in R_b(\mathbf{x})\}}}{\#\{j : \mathbf{X}_j \in R_b(\mathbf{x})\}}.$$

Averaging over B trees, the weights are

$$v_i(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} v_{i,b}(\mathbf{x}),$$

which sum to one. The survival function can then be constructed by using a weighted version of the NPMLE. Since the weights can be viewed as replications of the corresponding observations, the corresponding log likelihood function to be maximized can be written as

$$\log L(S(\cdot)|\mathcal{Y}, \mathbf{x}) = \log\left[\prod_{i=1}^{n} \mathbb{P}(L_i < T_i \leq R_i)^{v_i(\mathbf{x})}\right].$$

In practice, such an estimator can be constructed using the algorithm proposed by Turnbull (1976). Denote the Turnbull intervals as $\mathcal{I} = \{(\tau_{11}, \tau_{12}], (\tau_{21}, \tau_{22}], ..., (\tau_{l1}, \tau_{l2}]\}$ and the mass that is assigned to $(\tau_{j1}, \tau_{j2}]$ as $u_j = \mathbb{P}(\tau_{j1} < T \leq \tau_{j2}) = S(\tau_{j1}) - S(\tau_{j2})$, for $j = 1, 2, ..., l$. Maximization of $\log L(S(\cdot)|\mathcal{Y}, \mathbf{x})$ reduces to maximization of the following log likelihood function:

$$\log L_T(u_1, ..., u_l|\mathbf{x}) = \log\left[\prod_{i=1}^{n}\left(\sum_{j=1}^{l}\alpha_j^i u_j\right)^{v_i(\mathbf{x})}\right] = \sum_{i=1}^{n} v_i(\mathbf{x})\left(\log\sum_{j=1}^{l}\alpha_j^i u_j\right), \tag{2.1}$$

where $\alpha_j^i = \mathbb{I}\{(\tau_{j-1}, \tau_j] \subseteq (L_i, R_i]\}$ and the parameters are subject to the constraints $u_j \geq 0$ and $\sum_{j=1}^{l} u_j = 1$. Since the weights $v_1(\mathbf{x}), ..., v_n(\mathbf{x})$ define the forest-based adaptive neighborhood of $\mathbf{x}$, the resulting estimator from the weighting scheme can be viewed as a locally adaptive maximum likelihood estimator.

The weighted version of Turnbull's self-consistent estimator of $(u_1, u_2, ..., u_l)$ can be obtained as the solution of the simultaneous equation

$$\widehat{u}_j(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} v_i(\mathbf{x})\frac{\alpha_j^i}{\sum_{k=1}^{l}\alpha_k^i\widehat{u}_k}\widehat{u}_j(\mathbf{x}), \quad 1 \leq j \leq l.$$

Turnbull's estimator uses a self-consistency argument to motivate an iterative algorithm for the NPMLE, which turns out to be a special case of the EM-algorithm. Anderson-Bergman (2017) recently proposed an efficient implementation of the EMICM algorithm to fit the NPMLE, which greatly improves the computation power and therefore enables efficient prediction from the forest for interval-censored data. In the case of weighted observations, the EM step uses the same log likelihood function as in (2.1), and the ICM step, which reparameterizes the problem in terms of the vector $\Lambda_{jk} = \log(-\log S(\tau_{jk}))$ for $k = 1, 2$, $j = 1, 2, ..., l$, is to update the likelihood function as

$$\sum_{i=1}^{n} v_i(\mathbf{x})\left(\log\sum_{j=1}^{l}\alpha_j^i[\exp(-\exp(\Lambda_{j1})) - \exp(-\exp(\Lambda_{j2}))]\right).$$

This is then approximated with a second-order Taylor expansion for maximization (Anderson-Bergman, 2017).

2.2. *Regulating the construction of the IC ctrees in the IC cforest*

As discussed in Section 2.1, only a random subset of covariates is considered for splitting at each node. The size of this random set is denoted by *mtry*. It will be shown later that *mtry* is a very important tuning parameter. Other parameters such as *minsplit* (the minimum sum of weights in a node in order to be considered for splitting), *minprob* (the minimum proportion of observations needed to establish a terminal node), and *minbucket* (the minimum sum of weights in a terminal node), which control whether or not to implement a split (and thereby regulate the size of the individual trees), can potentially be essential in avoiding overfitting, and therefore may improve the overall performance.

The recommended values for these parameters are usually given as defaults to the algorithm. For example, *mtry* is usually set to be $\sqrt{m}$, where m is the number of covariates (Hothorn *and others*, 2006a; Ishwaran *and others*, 2008). However, in practice, we find that the choice of these parameters has a non-negligible effect on the overall performance of the proposed ensemble method. Hastie *and others* (2001) suggest that the best values for these parameters depend on the problem and they should be treated as tuning parameters. How these parameters affect the performance of proposed IC cforest and further guidelines on how to set these values are discussed in Section 3.3.

2.3. *Other ensemble resampling methods*

Recently, two papers introduced novel approaches to constructing ensemble methods for survival data. Steingrimsson *and others* (2018) proposed censoring unbiased regression survival trees and ensembles by extending the theory of censoring unbiased transformations applicable to loss functions for right-censored survival data. This new class of ensemble algorithms extends the RSF algorithm for use with an arbitrary loss function and allows the use of more general bootstrap procedures, such as the exchangeably weighted bootstrap (Weng, 1989). The extension of the theory of censoring unbiased transformation is not applicable in our context since the conditional inference framework uses multiple testing procedures that measure the association between responses and covariates for variable selection and splitting procedure, rather than loss minimization. The exchangeably weighted bootstrap procedures, including Bayesian bootstrap and the iid weighted bootstrap with weights simulated from a Gamma distribution, assign strictly positive real-valued weights to each observation in every bootstrap sample. This is in contrast to the nonparametric bootstrap approach that is used in the conditional inference forest framework, which places positive integer weights that sum to the sample size on approximately 63% of the observations in any given bootstrap sample. With these weights the exchangeably weighted bootstrap can avoid generating additional ties in the response variable when it is applied to censored survival data. Unfortunately, it is computationally infeasible in the conditional inference forest framework because resampling using the real-valued weights would require algorithm weights that effectively make the sample size orders of magnitude larger.

Wang and Zhou (2017) developed a RSF with space extension algorithm by combining random subspace, bagging, and extended space techniques. The extended covariate space used for model building contains all of the original covariates plus new covariates formed by differencing two randomly selected original ones. It can be applied to aggregation schemes that average predictions, as is done in the RSF, but is inapplicable to aggregation schemes that average observation weights, as is done in the conditional inference forest. This is because when using extended space techniques the covariates of each observation change for each bootstrap base learner replication. For these reasons, we will discuss only the standard conditional inference forest construction in this article.

3. PROPERTIES OF THE CONDITIONAL INFERENCE FOREST METHOD

In this section, we use computer simulations to investigate the properties of the proposed IC cforest estimation method. The event time T is generated from distribution $F(t)$ and the gap δ_t between any two

consecutive examination times from a distribution $G(t)$. The jth of in total $k+1$ examination times therefore is $t_j = \sum_{i=1}^{j} \delta_i$ and the intervals will be $(0, t_1], (t_1, t_2], ..., (t_k, \infty)$, each with width $\delta_i, i = 1, ..., k+1$. The censoring interval of T is the one that contains T. Here $F(t)$ and $G(t)$ are independent, and therefore the survival times T and the censoring mechanism are independent. This mechanism ensures the possibility that some observations can potentially be right-censored, that is T lies in (t_k, ∞).

We will study the properties of the proposed cforest method in terms of its estimation performance. The simulation setups are similar to those in Fu and Simonoff (2017).

3.1. *Model setup*

We use three simulation setups, each with five distributions ($F(t)$) of survival (event) time T to test the prediction performance of the proposed IC cforest.

In the first setup, the underlying true model has a tree structure. There are ten covariates $X_1, ..., X_{10}$, where X_1, X_4, and X_7 randomly take values from the set $\{1, 2, 3, 4, 5\}$, X_2, X_5, and X_8 are binary $\{1, 2\}$ and X_3, X_6, X_9, X_{10} are $U[0, 2]$. Only the first three covariates X_1, X_2, X_3 determine the distribution of the survival (event) time T. The survival time T follows distribution $\widetilde{T}_1$, $\widetilde{T}_2$, $\widetilde{T}_3$, or $\widetilde{T}_4$ according to the values of X_1, X_2, X_3 by a tree structure as follows:

- When $X_1 \leq 2$
  - if $X_2 \leq 1$, distributed as $\widetilde{T}_1$.
  - if $X_2 > 1$, distributed as $\widetilde{T}_2$.
- When $X_1 > 2$
  - if $X_3 \leq 1$, distributed as $\widetilde{T}_3$.
  - if $X_3 > 1$, distributed as $\widetilde{T}_4$.

The survival time T is generated from one of five different possible distributions (with each of the four (pairs of) parameter values corresponding to $\widetilde{T}_1$, $\widetilde{T}_2$, $\widetilde{T}_3$, and $\widetilde{T}_4$):

1. Exponential with four different values of λ from $\{0.1, 0.23, 0.4, 0.9\}$.
2. Weibull distribution with shape parameter $\alpha = 0.9$, which corresponds to decreasing hazard with time. The scale parameter β takes the values $\{7.0, 3.0, 2.5, 1.0\}$.
3. Weibull distribution with shape parameter $\alpha = 3$, which corresponds to increasing hazard with time. The scale parameter β takes the values $\{2.0, 4.3, 6.2, 10.0\}$.
4. Log-normal distribution with location parameter μ and scale parameter σ with four different pairs $(\mu, \sigma) = \{(2.0, 0.3), (1.7, 0.2), (1.3, 0.3), (0.5, 0.5)\}$.
5. Bathtub-shaped hazard model (Hjorth, 1980). The survival function is given by

$$S(t; a, b, c) = \frac{\exp\left(-\frac{1}{2}at^2\right)}{(1+ct)^{b/c}},$$

with $b = 1$, $c = 5$, and a set to take values $\{0.01, 0.15, 0.20, 0.90\}$.

The second and third setups are similar to those in Hothorn *and others* (2004),

- Second: Linear survival relationship with $\vartheta = -X_1 - X_2$.
- Third: Nonlinear survival relationship with $\vartheta = -\left[-\cos((X_1 + X_2) \cdot \pi) + \sqrt{X_1 + X_2}\right]$.

Here, ϑ is a location parameter whose value is determined by covariates X_1 and X_2. In these settings, six independent covariates $X_1, ..., X_{10}$ serve as predictor variables, with X_2, X_3, X_6, X_8, X_9 binary $\{0, 1\}$ and $X_1, X_4, X_5, X_7, X_{10}$ uniform $[0, 1]$. The survival time T_i again depends on ϑ with five different possible distributions:

1. Exponential with parameter $\lambda = e^{\vartheta}$;
2. Weibull with increasing hazard, scale parameter $\lambda = 10e^{\vartheta}$ and shape parameter $k = 2$;
3. Weibull with decreasing hazard, scale parameter $\lambda = 5e^{\vartheta}$ and shape parameter $k = 0.5$;
4. Log-normal distribution with location parameter $\mu = 1.5$ and scale parameter $\sigma = e^{\vartheta}$;
5. Bathtub-shaped hazard model (Hjorth, 1980). The survival function is given by

$$S(t; a, b, c) = \frac{\exp\left(-\frac{1}{2}at^2\right)}{(1 + ct)^{b/c}},$$

with $b = 1$, $c = 5$, and $a = e^{\vartheta}$.

To see how the IC cforest compares with a (semi-)parametric model and the corresponding tree model, we also include the Cox proportional hazards model implemented in the R package icenReg (Anderson-Bergman, 2016) (we will refer to this as IC Cox) and the IC ctree model implemented in the R package LTRCtrees (Fu and Simonoff, 2018) in the simulations for comparison. To see the amount of information loss due to interval-censoring, the oracle versions of all three models, Cox, ctree, and cforest, which are fitted using the actual event time T, are also included as in Hothorn *and others* (2006b).

In the second setup where $\vartheta = -X_1 - X_2$, the linear proportional hazards assumption is satisfied, so the Cox PH model should perform best. The third setup is similar to the second except that ϑ in this setup has a more complex nonlinear structure in terms of covariates, which is potentially more like a real-world application. This complex structure can make the distributions of T_i satisfy neither the Cox PH model nor the tree structure.

In all three simulation setups with five distributions $F(t)$, we consider three different distributions $G(t)$ of censoring interval width $\delta_t = t_{j+1} - t_j$,

1. $G_1(t)$, Uniform distribution $U(0.15, 0.35)$;
2. $G_2(t)$, Uniform distribution $U(0.75, 0.95)$;
3. $G_3(t)$, Uniform distribution $U(1.65, 1.85)$.

Notice that censoring interval widths generated by $G_2(t)$ should be around three times wider than those generated by $G_1(t)$, and censoring interval widths generated by $G_3(t)$ should be around seven times wider than those generated by $G_1(t)$. Intuitively, as the width of the censoring interval gets wider, less information about the actual survival time is available.

We also consider three possible right-censoring rates, 0% right-censoring, light censoring with about 20% observations being right-censored, and heavy censoring with about 40% observations being right-censored.

The simulation setup is designed to investigate the extent to which estimation performance of the proposed IC cforest deteriorates with the loss of information due to widening of censoring intervals, and also due to the increasing rate of right censoring.

3.2. *Evaluation methods*

To evaluate estimation performance, the average integrated L_2 distance between the true and the estimated survival curves

$$\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\max_j(T_j)}\int_0^{\max_j(T_j)}[\hat{S}_i(t)-S_i(t)]^2\mathrm{d}t \tag{3.2}$$

is used, where T_j is the (actual) event time of the jth observation and $\widehat{S}_i(\cdot)$ ($S_i(\cdot)$) is the estimated (true) survival function for the ith observation from a particular estimator.

3.3. *Evaluation of tuning parameters*

3.3.1. *mtry as a tuning parameter* In the cforest algorithm, a random selection of *mtry* input variables is used in each node for each tree. A split is established when all of the following criteria are met: (i) the sum of the weights in the current node is larger than *minsplit*, (ii) a fraction of the sum of weights of more than *minprob* will be contained in all daughter nodes, (iii) the sum of the weights in all daughter nodes exceeds *minbucket*, and (iv) the depth of the tree is smaller than *maxdepth*. Default values of *mtry*, *minsplit*, *minprob*, *minbucket,* and *maxdepth* have been given in *ctree_control* of the R package partykit (Hothorn *and others*, 2018), where *mtry* is set to be $\sqrt{m}$ (where m is the number of covariates), and the other four parameters are set to be {20, 14, 7, Inf}. Since typically unstopped and unpruned trees are used in random forests, we do not see *maxdepth* as a tuning parameter in the proposed IC cforest method.

The value of *mtry* can be fined-tuned on the "out-of-bag observations." The "out-of-bag observations" for the bth tree are those observations that are left out of the bth bootstrap sample and not used in the construction of the bth tree (in fact, about one-third of the observations in the original sample are "out-of-bag observations" for each bootstrap sample). The response for the ith observation can then be predicted by using each of the B trees in which that observation was "out-of-bag" (this will yield around $B/3$ predictions for the ith observation). The resulting prediction error is a valid estimate of the test error for the ensemble method. The idea of tuning *mtry* on the out-of-bag observations is borrowed from the function `tuneRF()` in the R package randomForest (Breiman *and others*, 2018). A version of `tuneRF()` for interval-censored data starts with the default values of *mtry*, then searches for the optimal values with a prespecified step factor with respect to out-of-bag error estimate *mtry* for IC cforest. The integrated Brier score (Graf *and others*, 1999), which is the most popular measure of prediction error in survival analysis, is used in the function `tuneRF()` for right-censored time data. Tsouprou (2015) adapted the integrated Brier score (IBS) to interval-censored time data,

$$IBS = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{T_{\max}}\int_0^{T_{\max}}[\mathbb{I}(T_i>t)-\widehat{S}_i(t)]^2\mathrm{d}t \tag{3.3}$$

with $T_{\max}=\max_{i=1,\ldots,n}\{L_i,R_i\}$ and $\mathbb{I}(T_i>t)$ estimated by

$$\widehat{\mathbb{I}}(T_i>t)=\frac{\widehat{S}_i(t)-\widehat{S}_i(R_i)}{\widehat{S}_i(L_i)-\widehat{S}_i(R_i)},$$

where $\widehat{S}_i(\cdot)$ is the estimated survival function for the ith observation. Using this evaluation measure, we can tune the *mtry* by the "out-of-bag" tuning procedure in Algorithm 1.

Algorithm 1 "Out-of-bag" tuning procedure for *mtry*

procedure tuneICCF($\{x, L, R\}_{i=1}^n$, stepFactor)
  $s \leftarrow$ stepFactor
  $r_1 \leftarrow \min\{r \in \mathbb{N}; \sqrt{m}/s^r > 1\}$
  $r_2 \leftarrow \max\{r \in \mathbb{N}; \sqrt{m}s^r < m\}$
  $mtrypool \leftarrow \{1, \sqrt{m}/s^{r_1}, \sqrt{m}/s^{r_1-1}, ..., \sqrt{m}s^{r_2-1}, \sqrt{m}s^{r_2}, m\}$
  for *mtry* in *mtrypool* **do**
    iccf.obj $\leftarrow$ ICcforest(data $= \{x, L, R\}_{i=1}^n$, mtryTest = *mtry*)
    pred.oob $\leftarrow$ predict(iccf.obj, OOB = TRUE)
    err.oob $\leftarrow$ sbrier_IC($\{x, L, R\}_{i=1}^n$, pred.oob) ▷ calculating IBS defined in (3.3)
  end
  $i^* \leftarrow \arg\min$ err.oob
  $mtry^* \leftarrow mtrypool[i^*]$
return $mtry^*$.

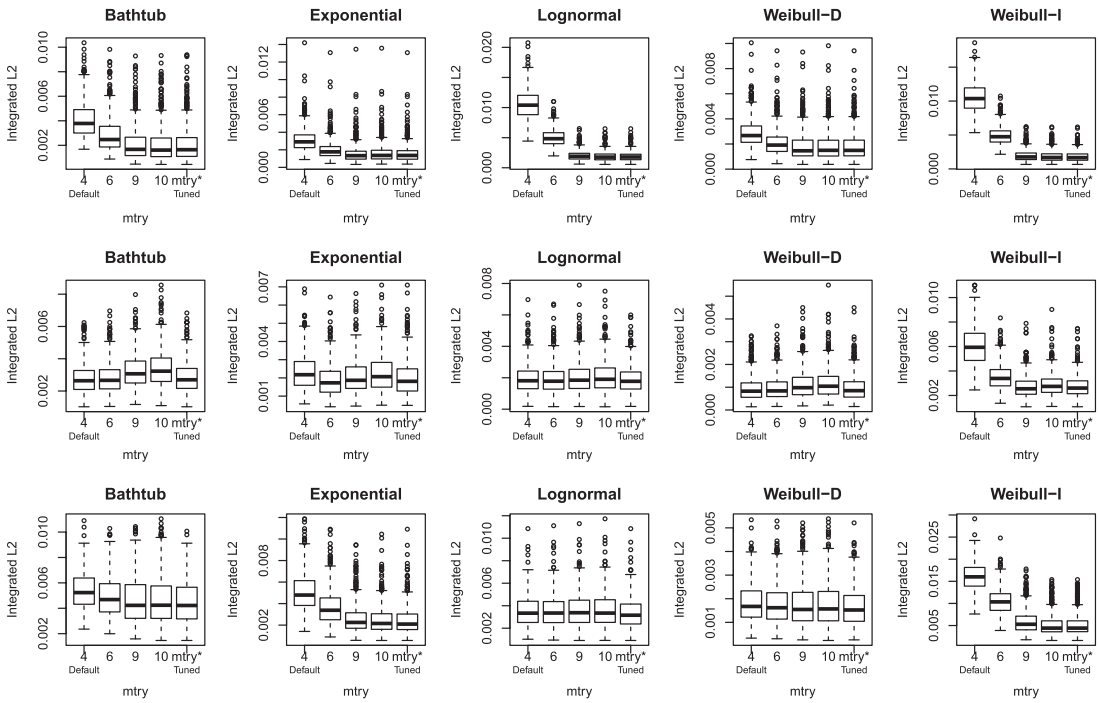

Fig. 1. Integrated L_2 difference of IC cforest with different *mtry* values, with $n = 200$, no right censoring and the interval censoring width generated by $G_1(t)$. The default value in `cforest` function is $\sqrt{m} \approx 4$. The value of *mtry* tuned by the "out-of-bag" tuning procedure is given in the last column in each boxplot. Top row gives results for the first setup (tree structure), middle row for the second setup (linear model), and bottom row for the third setup (nonlinear model).

Figure 1 gives an example of how IC cforest performs with different values of *mtry*. The *mtry* values are chosen using *stepFactor* $s = 1.5$ in Algorithm 1. In this example, the default value of *mtry* in the `cforest` function is not always optimal and sometimes the performance can be significantly improved by setting a larger value (values smaller than the default value never had better performance, so they are

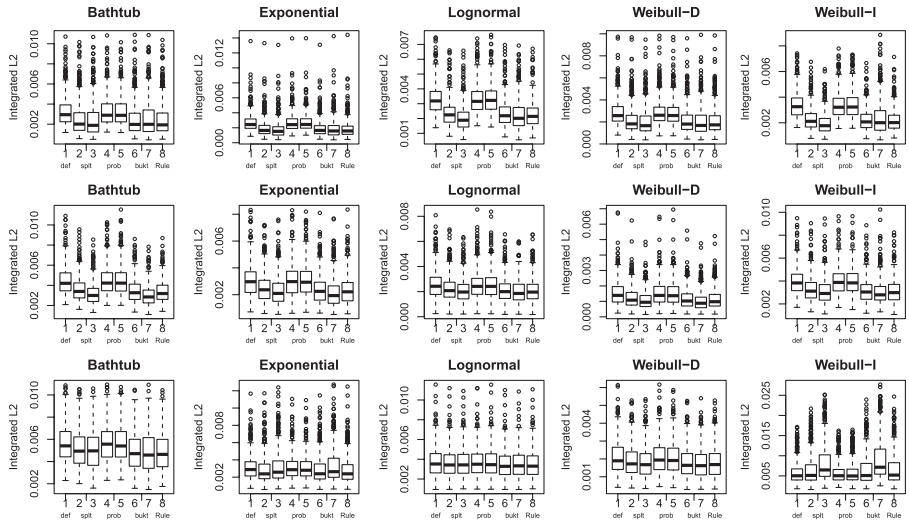

Fig. 2. Integrated L_2 difference of IC cforest with different *minsplit*, *minprob,* and *minbucket* values, with $n = 200$, no right-censoring and the interval-censoring width generated by $G_1(t)$. 1—(*minsplit*, *minprob*, *minbucket*) $= (20, 0.01, 7)$. 2—(*minsplit*, *minprob*, *minbucket*) $= (30, 0.01, 7)$, 3—(*minsplit*, *minprob*, *minbucket*) $= (40, 0.01, 7)$, 4—(*minsplit*, *minprob*, *minbucket*) $= (20, 0.05, 7)$, 5—(*minsplit*, *minprob*, *minbucket*) $= (20, 0.10, 7)$, 6—(*minsplit*, *minprob*, *minbucket*) $= (20, 0.01, 12)$, 7—(*minsplit*, *minprob*, *minbucket*) $= (20, 0.01, 16)$, 8—the "15%-Default-6% Rule": (*minsplit*, *minprob*, *minbucket*) $= (30, 0.01, 12)$. Top row gives results for the first setup (tree structure), middle row for the second setup (linear model), and bottom row for the third setup (nonlinear model).

not given). In fact, different distributions with different underlying models favor different values of *mtry*. The "out-of-bag" tuning procedure provides a relatively reliable choice of *mtry* that gives relatively good performance overall.

The size $n = 200$ with no right censoring and the censoring interval width generated by $G_1(t)$ is used in the simulations presented in Figure 1; results with $n = 500$ and $n = 1000$ were similar and are given in Section A.1 and Section B.1 of the supplementary material available at *Biostatistics* online, respectively.

3.3.2. *minsplit, minprob, and minbucket as tuning parameters* The optimal values that determine the split vary from case to case. As a fixed number, the default values may not affect the splitting at all when the sample size is large, while having a noticeable effect in smaller data sets. This inconsistency can potentially result in good performance in some data sets and poor performance in others. Here, we wish to determine a rule that can automatically adjust those values to the size of the data set, whose performance is relatively stable and better than that of the default values.

The values of *minsplit*, *minprob,* and *minbucket* determine whether a split in a node will be implemented. We design our experiments to explore the individual effect of each parameter. Based on the results, we propose the "15%-Default-6% Rule," which is to set *minsplit* to be 15% of the sample size *n*, *minprob* to be the default value, and *minbucket* to be 6% of the sample size *n*.

Figure 2 gives an example of the sensitivity of IC cforest to the different values of *minsplit*, *minprob*, and *minbucket*. The choices of *minsplit* are 20 (default value), 30 (15% of the sample size *n*), and 40 (20% of the sample size *n*). The choices of *minprob* are 0.01 (default value), 0.05, and 0.10. The choices of *minbucket* are 7 (default value), 12 (6% of the sample size *n*), and 16 (8% of the sample size *n*). In each plot of Figure 2, column 1 shows the integrated L_2 under the default setting, columns 2–7 show the integrated L_2 differences when changing the value of one parameter at a time while holding the others the same, and

column 8 shows the results of the proposed "15%-Default-6% Rule." Here the performance of IC cforest is shown with a limited number of values and these values are selected to give as much understanding of the performance change due to the tuning parameters as possible. We can see that overall the value of *minprob* does not change the performance much (as expected, since we set the equivalent parameter, *minbucket*, to be a much larger proportion of the size of the data set), while changing *minsplit* and *minbucket* can possibly improve the performance of the overall performance. Empirically, the "15%-Default-6% Rule" has shown to improve the overall performance over the default setting under different models with different distributions. The simulation results show that a slightly larger size of leaf is favored, since the smaller default size makes the forest more prone to capturing noise and overfitting, and therefore exhibits worse performance.

The size $n = 200$ with no right censoring and the censoring interval width generated by $G_1(t)$ is used in the simulations presented here; results with $n = 500$ and $n = 1000$ were similar and are given in Sections A.2 and B.2 of the supplementary material available at *Biostatistics* online, respectively.

3.4. *Estimation performance*

We run 500 simulation trials for each setting to see how well the proposed IC cforest performs compared to the IC Cox model and the corresponding IC ctree model. The parameter *mtry* in IC cforest is tuned following the "out-of-bag" tuning procedure and the values for *minsplit*, *minprob,* and *minbucket* are chosen using the "15%-Default-6% Rule" described in Section 3.3. The size $n = 200$ with censoring interval width generated by $G_1(t)$ and light right-censoring rate (20%) is used in the simulations presented here; results with no (0%) or heavy right-censoring rate (60%) were similar and are given in Sections C.1 and C.2 of the supplementary material available at *Biostatistics* online, respectively; results with $n = 500$ and $n = 1000$ were also similar and are given in Sections D and E of the supplementary material available at *Biostatistics* online, respectively.

Figure 3 gives side-by-side integrated L_2 difference boxplots for all three setups with sample size $n = 200$ and with censoring width generated from $G_1(t)$ and with light right-censoring rate. We can see that the "out-of-bag" tuning procedure and the "15%-Default-6% Rule" improve the IC cforest performance over the parameters set by default. The top row in Figure 3 shows that in the presence of right-censoring, the proposed IC cforest performs as least as well as the IC ctree method in the first setup, where the true model is a tree. In addition, for all five distributions, the IC cforest outperforms the IC Cox model.

As expected, the IC Cox model can outperform the IC cforest method in the second setup (where the true model is a linear model). This occurs when the underlying distribution is the Weibull-Increasing distribution, but the proposed IC cforest can generally represent a linear model as well as the IC Cox model or even better than it.

IC ctree outperforms IC Cox model in the third setup due to its flexible structure (Fu and Simonoff, 2017), and we can see from the bottom row in Figure 3, that the proposed IC cforest further improves the performance and shows its advantage in a relatively complex survival relationship.

The censoring interval width generating distribution $G_1(t)$ is used in the simulations presented here. Intuitively, a wider censoring interval, meaning less information and more uncertainty, will result in poorer performance in the forest.

How the censoring interval width affects the performance of IC cforest has also been investigated. When the censoring interval width is small, IC cforest can perform as well as the "Oracle," where the true survival times are known, and there is no right-censoring. When the censoring interval width is roughly three times wider, the loss of information starts to affect the IC cforest performance, but not greatly. When the censoring interval width is roughly seven times wider, the IC cforest performance deteriorates considerably more. The illustration figure is given in Section C.3 of the supplementary material available at *Biostatistics* online.

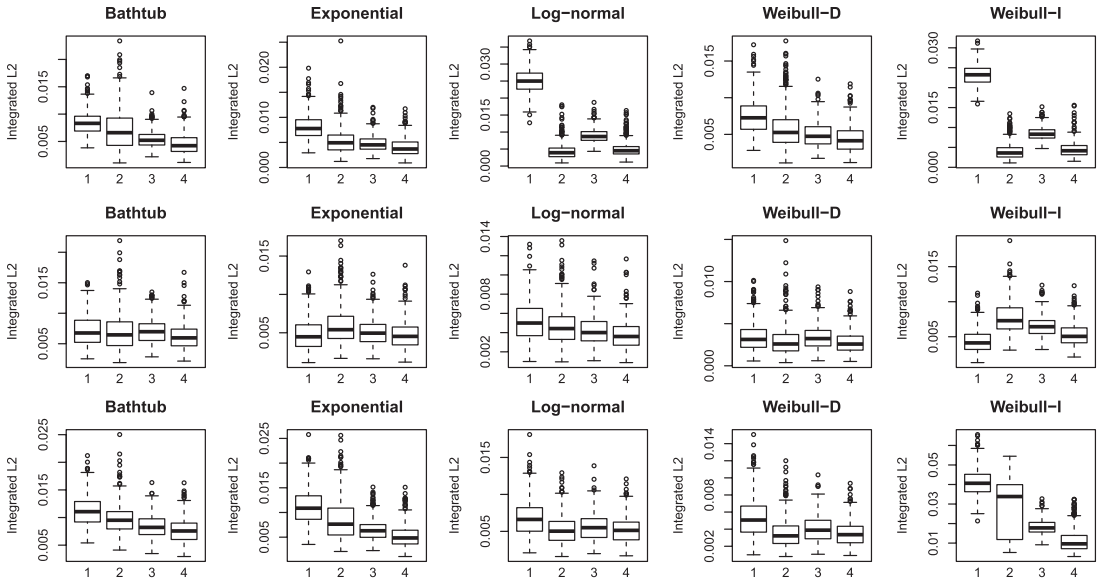

Fig. 3. Integrated L_2 difference with $n = 200$, censoring interval width generated from $G_1(t)$, with light (right-) censoring rate. Methods are numbered as 1—IC Cox model, 2—IC ctree, 3—IC cforest and parameters set by default, and 4—IC cforest with parameters set through "out-of-bag" tuning procedure and the "15%-Default-6% Rule." Top row gives results for the first setup (tree structure), middle row for the second setup (linear model), and bottom row for the third setup (nonlinear model).

In fact, this loss of information due to the increased censoring interval widths affects all three different methods, and the patterns across methods we have seen in Figures 3 with censoring interval width generating distribution $G_1(t)$ are similar to those with $G_2(t)$ and $G_3(t)$. That is, the proposed IC cforest can still outperform the IC ctree method even under the tree model and outperform the IC Cox model under a linear model. For example, the patterns across the three methods for each model preserve well under the change of censoring interval widths in the situation with no right-censoring. The illustration figure is given in Section C.4 of the supplementary material available at *Biostatistics* online.

4. Real data set

The Signal Tandmobiel® study is a longitudinal prospective oral health study that was conducted in the Flanders region of Belgium from 1996 to 2001. In this study, 4430 first year primary school schoolchildren were randomly sampled at the beginning of the study and were dental-examined annually by trained dentists. The data consist of at most six dental observations for each child including time of tooth emergence, caries experience, and data on dietary and oral hygiene habits. The details of study design and research methodology can be found in Vanobbergen *and others* (2000). The data are provided as the tandmob2 data set in the R package bayesSurv (Komárek, 2015). The tandmob2 data set provides the time to emergence of 28 teeth in total. Each of the tooth emergence times can be taken as a response variable and we can test the prediction performance of the proposed IC cforest method, compared to the corresponding IC ctree method and IC Cox method. Potential predictors of emergence time of the child's tooth include gender, province, evidence of fluoride intake, type of educational system, starting age of brushing teeth, whether each of the 12 deciduous teeth were decayed or missing due to caries or filled, whether each of the 12

deciduous teeth were removed because of orthodontic reasons, and whether each of the 12 deciduous teeth were removed due to the orthodontic reasons or decayed on at most the last examination before the first examination when the emergence of the permanent successor was recorded. These potential predictors cover all of the variables in the data set.

To compare different methods, we conducted leave-one-out cross-validation on the entire data set, and then computed the average absolute prediction distance below L_i or above R_i when the predicted median emergence time falls outside of the observed interval, which measures the distance away from the interval for those observations (if a predicted emergence time falls within the observed emergence interval it is impossible to say what the prediction error is, so such observations are not considered).

The IC cforest method applied with *mtry* chosen through the "out-of-bag" tuning procedure and *minplit*, *minprob*, *minbucket* chosen by the "15%-Default-6% Rule," IC ctree, and the IC Cox model are applied to each of the tooth data sets. Table 1 shows that the proportion of the time the predicted median emergence

Table 1. *Evaluation on 28 tooth data sets in Signal Tandmobiel® Study. In the table, p_{out} denotes the proportion of the predicted median emergence times lying outside censoring intervals, $\bar{d}_{out}$ denotes average absolute prediction distance below L_i or above R_i, and the bolded value in each row indicates the smallest one among the three $\bar{d}_{out}$'s.*

Tooth	IC Cox		IC ctree		IC cforest	
	p_{out}(%)	$\bar{d}_{\text{out}}$	p_{out}(%)	$\bar{d}_{\text{out}}$	p_{out}(%)	$\bar{d}_{\text{out}}$
11	33.7	0.3558	33.0	**0.3489**	32.1	0.3732
21	34.2	**0.3428**	33.2	0.3439	33.7	0.3639
31	23.6	84.1325	21.5	**0.3195**	20.9	0.3312
41	21.4	71.1985	17.4	0.6236	18.0	**0.6019**
12	54.0	0.5259	52.6	0.5369	54.3	**0.5187**
22	51.0	0.5215	50.3	0.5232	52.1	**0.5026**
32	38.1	0.4036	37.4	0.4050	37.7	**0.4010**
42	39.4	0.4004	38.1	0.4110	39.5	**0.3969**
13	57.8	0.6894	57.6	**0.6236**	56.7	0.6564
23	59.1	1.3304	60.6	0.5863	60.1	**0.5822**
33	64.4	0.6454	71.3	**0.6279**	65.6	0.6926
43	63.6	0.6386	63.6	0.6434	64.6	**0.6304**
14	66.8	**0.7239**	65.6	0.7479	67.0	0.7311
24	67.0	0.7082	68.0	**0.6934**	66.8	0.7176
34	66.1	**0.6976**	66.4	0.7012	66.3	0.7109
44	65.0	0.7108	65.8	**0.7022**	66.6	0.7221
15	55.6	0.7141	58.7	0.6602	56.4	**0.6382**
25	55.9	2.0519	60.1	0.6635	58.5	**0.6629**
35	52.6	0.7245	56.6	0.6670	55.9	**0.6401**
45	51.5	0.7221	52.4	0.6866	54.7	**0.6374**
16	25.5	**0.3138**	22.0	0.3765	23.3	0.3470
26	26.4	0.3250	22.8	0.3300	22.8	**0.3237**
36	27.5	0.4036	28.0	**0.3274**	27.0	0.3304
46	26.6	**0.3125**	24.1	0.3277	24.3	0.3234
17	28.8	55.2018	28.5	28.0678	28.0	**11.4780**
27	30.6	96.5333	31.3	43.3953	30.9	**30.2143**
37	46.3	0.5876	48.2	**0.5157**	47.2	0.5436
47	43.1	6.1757	46.3	**0.5615**	43.7	0.5935

falls outside the observed intervals is roughly the same for the three methods, although it varies greatly from tooth to tooth. Among these 28 tooth data sets IC cforest gives the smallest average absolute prediction distance away from the observed intervals for those observations that fall outside of them for 50% of the teeth; the IC ctree follows (32%) and the IC Cox model trails both (18%). Thus, the IC cforest method does a good job of predicting the actual emergence times.

5. Conclusion

In this article, we have proposed a new ensemble algorithm based on the conditional inference survival forest designed to handle interval-censored data. Through the use of a simulation study, we see that the proposed IC cforest method can outperform the IC ctree and the IC Cox proportional hazards model even when the underlying true model is designed for the tree structure or the linear relationship, respectively, in terms of prediction performance, and clearly outperforms both in the nonlinear situation that neither is designed for.

The tuning parameters in the proposed IC cforest affect the overall performance of the method. In this article, we have provided guidance on how to choose those parameters to improve on the potentially poor performance of the default settings. Further investigation of the best way to choose these parameters in a data-dependent way would be useful. It would also be interesting to extend these results to competing risks data.

6. Software

An R package, ICcforest, that implements the IC cforest method is available at CRAN. R scripts for reproducibility of the illustrative example analysis are available from https://github.com/ElainaYao/ICdata_ICcforest.

7. Supplementary material

Supplementary material is available online at http://biostatistics.oxfordjournals.org.

Acknowledgments

We thank the referees for their helpful comments.

Conflict of Interest: None declared.

Funding

Data collection of the Signal Tandmobiel® data was supported by Unilever, Belgium. The Signal-Tandmobiel project comprises the following partners: Dominique Declerck (Department of Oral Health Sciences, KU Leuven), Luc Martens (Dental School, Gent Universiteit), Jackie Vanobbergen (Oral Health Promotion and Prevention, Flemish Dental Association and Dental School, Gent Universiteit), Peter Bottenberg (Dental School, Vrije Universiteit Brussel), Emmanuel Lesaffre (L-Biostat, KU Leuven), and Karel Hoppenbrouwers (Youth Health Department, KU Leuven; Flemish Association for Youth Health Care).

References

Anderson-Bergman, C. (2016). *icenReg: Regression Models for Interval Censored Data. Version 2.0.8.* https://CRAN.R-project.org/package=icenReg.

Anderson-Bergman, C. (2017). An efficient implementation of the EMICM algorithm for the interval censored NPMLE. *Journal of Computational and Graphical Statistics* **26**, 463–467.

ATHEY, S., TIBSHIRANI, J. AND WAGER, S. (2019). Generalized random forests. *The Annals of Statistics* **47**, 1148–1178.

BOGAERTS, K., KOMÁREK, A. AND LESAFFRE, E. (2017). *Survival Analysis with Interval-Censored Data: A Practical Approach with Examples in R, SAS and BUGS*. Boca Raton, FL: Chapman and Hall/CRC.

BREIMAN, L. (2001). Random forests. *Machine Learning* **45**, 5–22.

BREIMAN, L., CUTLER, A., LIAW, A. AND WIENER, M. (2018). *randomForest: Breiman and Cutler's Random Forests for Classification and Regression. Version 4.6-14*. https://CRAN.R-project.org/package=randomForest.

FINKELSTEIN, D. M. (1986). A proportional hazards model for interval-censored failure time data. *Biometrics* **42**, 845–854.

FU, W. AND SIMONOFF, J. S. (2017). Survival trees for interval-censored survival data. *Statistics in Medicine* **36**, 4831–4842.

FU, W. AND SIMONOFF, J. S. (2018). *LTRCtrees: Survival Trees to Fit Left-Truncated and Right-Censored and Interval-Censored Survival Data. Version 1.1.0.* https://CRAN.R-project.org/package=LTRCtrees.

GRAF, E., SCHMOOR, C., SAUERBREI, W. AND SCHUMACHER, M. (1999). Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine* **18**, 2529–2545.

HASTIE, T., TIBSHIRANI, R. AND FRIEDMAN, J. (2001). *The Elements of Statistical Learning*, Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

HOTHORN, T., BÜHLMANN, P., DUDOIT, S., MOLINARO, A. AND VAN DER LAAN, M. J. (2006a). Survival ensembles. *Biostatistics* **7**, 355–373.

HOTHORN, T., HORNIK, K. AND ZEILEIS, A. (2006b). Unbiased recursive partitioning: a conditional inference framework. *Journal of Computational and Graphical Statistics* **15**, 651–674.

HOTHORN, T., LAUSEN, B., BENNER, A. AND RADESPIEL-TRÖGER, M. (2004). Bagging survival trees. *Statistics in Medicine* **23**, 77–91.

HOTHORN, T., SEIBOLD, H. AND ZEILEIS, A. (2018). *partykit: A Toolkit with Infrastructure for Representing, Summarizing, and Visualizing Tree-Structured Regression and Classification Models. Version 1.2-2.* https://CRAN.R-project.org/package=partykit.

ISHWARAN, H., KOGALUR, U. B., BLACKSTONE, E. H. AND LAUER, M. S. (2008). Random survival forest. *The Annals of Applied Statistics* **2**, 841–860.

KOMÁREK, A. (2015). *bayesSurv: Bayesian Survival Regression with Flexible Error and Random Effects Distributions. Version 2.6.* https://CRAN.R-project.org/package=bayesSurv.

LIN, Y. AND JEON, Y. (2006). Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association* **101**, 578–590.

MEINSHAUSEN, N. (2006). Quantile regression forests. *The Journal of Machine Learning Research* **7**, 983–999.

PAN, W. (1998). Rank invariant tests with left truncated and interval censored data. *Journal of Statistical Computation and Simulation* **61**, 163–174.

STEINGRIMSSON, J. A., DIAO, L. AND L., STRAWDERMAN R. (2018). Censoring unbiased regression trees and ensembles. *Journal of the American Statistical Association* **114**, 370–383.

SUN, J. (2006). *The Statistical Analysis of Interval-Censored Failure Time Data*, Statistics for Biology and Health. New York, NY: Springer New York Inc.

TSOUPROU, S. (2015). Measures of discrimination and predictive accuracy for interval censored survival data, [Master's Thesis]. Leiden University.

TURNBULL, B. W. (1976). The empirical distribution function with arbitrarily grouped, censored and truncated data. *Journal of the Royal Statistical Society. Series B (Methodological)* **38**, 290–295.

VANOBBERGEN, J., MARTENS, L., LESAFFRE, E. AND DECLERCK, D. (2000). The Signal-Tandmobiel® project—a longitudinal intervention health promotion study in Flanders (Belgium): baseline and first year results. *European Journal of Paediatric Dentistry* **2**, 87–96.

WANG, H. AND ZHOU, L. (2017). Random survival forest with space extensions for censored data. *Artificial Intelligence in Medicine* **79**, 52–61.

WENG, C.-S. (1989). On a second-order asymptotic property of the Bayesian bootstrap mean. *The Annals of Statistics* **17**, 705–710.

[*Received January 26, 2019; revised June 11, 2019; accepted for publication June 14, 2019*]