The impact of churn on client value in health insurance, evaluation using a random forest under various censoring mechanisms

Guillaume Gerber, Yohann Le Faou, Olivier Lopez, Michael Trupin

The impact of churn on client value in health insurance, evaluation using a random forest under various censoring mechanisms

Guillaume Gerber[a], Yohann Le Faou[b], Olivier Lopez[c], and Michael Trupin[d]

[a]*Forsides, 52 rue de la Victoire, 75009, Paris, France*

[b]*Forsides & Sorbonne Université, CNRS, Laboratoire de Probabilités, Statistique et Modélisation, LPSM, 4 place Jussieu, F-75005 Paris, France, mail : yohann.lefaou@courriel.upmc.fr*

[c]*Sorbonne Université, CNRS, Laboratoire de Probabilités, Statistique et Modélisation, LPSM, 4 place Jussieu, F-75005 Paris, France, mail: olivier.lopez@sorbonne-universite.fr*

[d]*Santiane, 38 avenue des Champs-Elysées, 75008, Paris, France*

Abstract

In the insurance broker market, commissions received by brokers are closely related to so-called "customer value": the longer a policyholder keeps their contract, the more profit there is for the company and therefore the broker. Hence, predicting the time at which a potential policyholder will surrender their contract is essential in order to optimize a commercial process and define a prospect scoring. In this paper, we propose a weighted random forest model to address this problem. Our model is designed to compensate for the impact of random censoring. We investigate different types of assumptions on the censoring, studying both the cases where it is independent or not from the covariates. We compare our approach with other standard methods which apply in our setting, using simulated and real data analysis. We show that our approach is very competitive in terms of quadratic error in addressing the given problem.

1 Introduction

In insurance brokerage, an important problem is to evaluate what is known as "prospect value". Roughly speaking, this represents the rentability of a potential policyholder. More effort could then be dedicated to attract a prospect with probable high profitability. A first approach to evaluate this prospect value is to predict how long a given current customer will keep their contract. To investigate this, brokers now have access to large databases of potentially relevant information. In the present paper, our aim is to discuss an extension of the *random forest* algorithm which takes into account specific details of this framework. Among these, a crucial issue is to deal with right-censoring, which is a common problem when dealing with duration variables. Due to the temporal phenomenon that we study, a significant number of observations are incomplete and require particular care to counterbalance the corresponding lack of information. The method we propose consists of an appropriate weighting of the observations to compensate for censoring. We discuss different approaches and compare them using simulation and real data analysis.

The random forest algorithm was first proposed in Breiman (2001). The underlying aim is to estimate $E[\phi(T)|X]$ with the help of bootstrap data augmentation and aggregation of regression trees. In our case, T is a right-censored variable, X a vector of covariates, and ϕ a known gain function, namely the value of a customer remaining a time T in the portfolio. In the presence of censoring Hothorn, Bühlmann, et al. (2006) proposed a generalization extending the random forest algorithm. Using the *inverse probability of censoring weights* (IPCW) approach, described in Van der Laan & Robins (2003), they introduce weights to compensate for censoring. Additionally, Steingrimsson et al. (2016, 2018) combined doubly robust estimators with survival trees. The novelty of our approach is to compute IPCW in situations where censoring depends on the covariates, as it was suggested in Molinaro et al. (2004), and we will discuss its performance compared to other weighting strategies. The IPCW technique is a very general one: once the weights have been determined, any regression technique can be generalized to this framework – see for example Koul et al. (1981) for linear regression, Goldberg & Kosorok (2017) for support vector machines, and Lopez et al. (2016) for *classification and regression trees* (CART). The difficulty in defining appropriate weights is in determining which identifiability assumption is reasonable given the situation. The weights may be considerably different if censoring is allowed to depend on covariates X (see e.g., Lopez et al. (2013)), or not (see e.g., Stute (2003)), and, as we will show on real data, this has important consequences on the results of the procedure.

Another modification of random forests to the presence of censoring is the *random sur-*

vival forest (RSF) of Ishwaran et al. (2008) (see also Ishwaran & Kogalur (2007)). In this procedure, a log-rank test is used to split the observations at each computational step of the the regression trees. This type of procedure for growing a survival tree was previously proposed in Ciampi et al. (1986) and Segal (1988). It was also studied in LeBlanc & Crowley (1993) and in Hothorn, Hornik, & Zeileis (2006) in the context of conditional inference trees. Moreover, Zhu & Kosorok (2012) have studied the impact of recursively fitted RSF models on prediction quality. LeBlanc & Crowley (1992) introduced the *relative risk tree* (RRT) algorithm, where a proportional hazard likelihood is used as the criterion at the splitting step. Forests of relative risk trees were investigated in Hothorn et al. (2004) and Ishwaran et al. (2004). The extension to left-truncation of this approach was considered in Fu & Simonoff (2016). Additional splitting criteria examined in the literature include the exponential log-likelihood in Davis & Anderson (1989), and splits determined through analysis of residuals of the Cox model (Ahn & Loh (1994)). More recently, Zhu (2013) and Zhu et al. (2015) proposed the *reinforcement learning trees* (RLT) algorithm which consists in fitting an embedded model at each step of the tree construction to improve the selection of the splitting variable, while Li & Bradic (2019) explored censored quantile regression using random forests.

While the theoretical properties of tree and forest-based learning algorithms are not fully understood yet, the consistency of RSF was first investigated in Ishwaran & Kogalur (2010) under the assumption that X takes a finite number of values. These properties were further studied in Cui, Zhu, & Kosorok (2017) and Cui, Zhu, Zhou, & Kosorok (2017) which provided a theoretical framework to consider the consistency of survival forests, and established consistency under specific conditions that include random splitting rules and splitting rules with marginal signal checking. Cui, Zhu, Zhou, & Kosorok (2017) also underlined the problem of non optimal split selection for usual survival tree methods. The method we study in this article does not suffer from such problem since it requires less assumptions. Finally, a review of the literature on survival trees can be found in Bou-Hamad et al. (2011).

The rest of the paper is organized as follows. In Section 2, we present the specific details of the types of observations we examine here, and describe the relevant random forest algorithm we have adapted to censoring. A simulation study is performed in Section 3, while the real-world behavior of our procedure is illustrated in Section 4, where we present an application of this weighted random forest to model the churn behavior of health insurance policy holders. Further results as well as explanations about the choice of the random forest parameters are provided in supplementary material.

2 Description of the method

2.1 The survival regression setting

To study the termination risk, we introduce the *lifetime* of a contract, denoted T. This duration is not directly observed due to the presence of right-censoring (which is a classical issue in survival analysis). Instead of observing T, we observe a pair of variables (Y, δ) defined as

$$Y = \min(T, C),$$
$$\delta = \mathbb{1}_{T \leq C}.$$

The need for the censoring variable C is due to the fact that some contracts may not have been terminated at the end of the observation period. Additionally, a vector of covariates $X \in \mathcal{X} \subset \mathbb{R}^p$ is available, in order to identify certain characteristics that may influence the termination time. Observations correspond to i.i.d. replicates $(Y_i, \delta_i, X_i)_{1 \leq i \leq n}$ of this set-up. For the sake of simplicity, we consider the case where T and C are continuous random variables. For any continuous random variable U, we will denote by $S_U(t) = P(U > t)$ its survival function.

Our aim is to estimate the function $f(x) = E[\phi(T)|X = x]$. In our application, ϕ is a pricing function that associates a unitary profit with a customer remaining a time T with the portfolio in question.

Since T is not directly observed because of censoring, classical estimators of f (random forests, as developed below, or any other regression estimator) are biased if no attempt is made to correct for this phenomenon. The procedure we propose is based on the inverse probability of censoring weighting (IPCW, see e.g., Van der Laan & Robins (2003)) which, through an appropriate weighting of the observations, aims to suppress this bias. This general method is described in Section 2.2 below.

2.2 Inverse probability of censoring weighting

2.2.1 Introduction

IPCW is a general principle which consists of correcting the bias introduced by censoring. Given a continuous random variable V (possibly multidimensional), the key task is to determine which weights should be put on an uncensored observation in order to retrieve the distribution of the variable of interest.

Proposition 1 (The IPCW principle) *Let* $\gamma = \begin{cases} 1 \text{ if } V \text{ is not censored} \\ 0 \text{ if } V \text{ is censored} \end{cases}$ *and* $V' = \gamma V$.

Let $p(v) = P(\gamma = 1 | V = v)$ *and assume that* $\forall v, p(v) > 0$.
Then for any function ψ,

$$E\left[W \cdot \psi(V')\right] = E\left[\psi(V)\right], \quad \textit{with} \quad W = \frac{\gamma}{p(V')}.$$

This result states that given a random variable V that is not always observable, it is still possible to estimate its distribution based on the observations $(V'_i, \gamma_i)_{i=1,..,n}$. This is done by attributing weights $\frac{\gamma_i}{p(V'_i)}$ to the observations, with $p(v)$ the probability of V being non-censored, given $V = v$.

In our setting, we can apply the IPCW routine to the vector (T, X). This results in the equality:

$$E\left[W \cdot \psi(Y, X)\right] = E\left[\psi(T, X)\right], \tag{1}$$

where $W = \frac{\delta}{p(Y,X)}$ and $p(t, x) = P(\delta = 1 | T = t, X = x) = P(t \leq C | T = t, X = x)$.

In the survival setting, it is generally impossible to infer values for $p(t, x)$ since – as it is well known – the identifiability of the model requires assumptions on the dependence between T and C that cannot be statistically verified (see Section 4.1 in Lagakos (1979)). However, with assumptions on the dependence between T and C, it is possible to compute $p(t, x)$.

Let **H1** and **H2** denote the following hypotheses:

$$\begin{aligned} \textbf{H1} &: P(t \leq C | X = x, T = t) = S_C(t), \\ \textbf{H2} &: P(t \leq C | X = x, T = t) = S_C(t | X = x), \end{aligned}$$

with $S_C(\cdot | X = x)$ the conditional distribution function of C given $X = x$. Sufficient conditions for these hypotheses to be satisfied are, respectively, $(T, X) \perp\!\!\!\perp C$ (**H1**) and $T \perp\!\!\!\perp C$ conditionally on X (**H2**).

Clearly **H2** is more general than **H1**, but requires estimation of a conditional survival function, which is more tricky. Hence, if the strongest assumption **H1** is reasonable, it may be interesting to use it instead of **H2**, in order to facilitate computation of the appropriate weights. The obvious drawback is that, if the censoring is dependent on the covariates, **H1** does not hold.

In what follows, we consider these two assumptions as separate cases, but both are handled simultaneously. Depending on the hypothesis we make, we note $W = \delta/S_C(Y)$ (**H1**) or $W = \delta/S_C(Y|X)$ (**H2**), so that in both cases equation (1) holds. We also note for all $i \in \{1, .., n\}$, $W_i = \delta_i/S_C(Y_i)$ (or $W_i = \delta_i/S_C(Y_i|X = X_i)$) the exact IPCW weights. However, the function S_C (resp. $S_C(\cdot|X)$) is unknown and we have to estimate it in order to compute the weights W.

2.2.2 Computation of the weights

Computation of the IPCW weights requires us to estimate the survival function of the censoring variable. In many applications, the roles of the variables T and C are symmetric, and the censoring variable needs to be studied using survival models. In our work, we have used three strategies to estimate S_C (resp. $S_C(\cdot|X)$):

- S_C estimated with the Kaplan-Meier (KM) estimator (Kaplan & Meier (1958)): $\hat{S}_C(t) = \prod_{Y_i \leq t} \left(1 - (1 - \delta_i)/\sum_{j=1}^{n} \mathbb{1}_{Y_j \geq Y_i}\right)$.
- $S_C(\cdot|X)$ estimated with the Cox model (Cox (1972)). This model assumes that the hazard rate has the form: $\lambda_C(t|X = x) = \lambda_0(t) \cdot e^{{}^t\beta \cdot x}$ with β a vector of coefficients, and for some random variable U, $\lambda_U = -S'_U/S_U$.
- $S_C(\cdot|X)$ estimated with the *random survival forest* (RSF) algorithm (Ishwaran et al. (2008)).

An alternative idea for estimating $S_C(\cdot|X)$ would be to use a kernel estimator such as the conditional Kaplan-Meier estimator of Beran (1981) and Dabrowska (1989). Nevertheless, this would rely on kernel smoothing whose behavior deteriorates when the dimension of X increases (i.e. $p > 3$) as noted in Lopez et al. (2013). This dimensional constraint is the reason for imposing a stronger assumption on the conditional distribution of the censoring as a feasible compromise.

Given estimators $\hat{S}_C$ (resp. $\hat{S}_C(\cdot|X)$) of S_C (resp. $S_C(\cdot|X)$), the weights $(\hat{W}_i)_{i=1,..n}$ are computed using the formula $\hat{W}_i = \delta_i/\hat{S}_C(Y_i)$ (resp. $\hat{W}_i = \delta_i/\hat{S}_C(Y_i|X_i)$). Therefore, each method used to estimate S_C leads to different IPCW weights, then to different estimators of the joint distribution of (T, X). In the following section, the weights $(\hat{W}_i)_{i=1,..n}$ refer to any collection of weights computed with one of the three methods.

2.3 A weighted random forest algorithm for the regression of right-censored data

Random forest is an ensemble learning algorithm which consists of an aggregation of elemental regression trees (base learners), see e.g., Breiman (2001) and Biau & Scornet (2016). A regression tree (see Breiman et al. (1984)) produces a partition of a dataset using successive binary splits based on values of the covariates. In each subset obtained (called a *leaf*), the observations correspond to a homogeneous group. The key behind group formation is the choice of the splitting criterion (e.g., least squares) which aims to reduce heterogeneity at each step.

The random forest algorithm is a combination of two methods: a *classification and regression tree* (CART) algorithm to compute each tree, and a bagging algorithm (bagging means *bootstrap aggregating*, see Breiman (1996)) to introduce randomness into partition formation by building bootstrap samples. Compared to a single regression tree approach, tree aggregation in random forests stabilizes results by making them less sensitive to changes in the database.

The rest of this section is devoted to adapting the random forest algorithm to the presence of right censoring. In Section 2.3.1, we propose a modification of the splitting criterion to work in our framework. Computation of predictions given a tree is shown in Section 2.3.2. An overall description of the algorithm is given in Section 2.3.3, and we discuss its parameters in Section 2.3.4.

2.3.1 Split selection

In the building of a binary tree, the main interest is in the way splits are determined. Here, we describe the split selection procedure used at each node of the tree, for the growing of one tree of the forest. This tree is built using a bootstrap sample of the initial data.

Let $\mathcal{D}_n$ be a list of indices in $\{1, \dots, n\}$ which represents a bootstrap sample of size n of the initial data, drawn uniformly with replacement. Given a set $B \subset \mathcal{X}$, let $n^{\mathcal{D}_n,w}(B) = \sum_{i\in\mathcal{D}_n} W_i \mathbb{1}_{X_i\in B}$ be the weighted number of observations of $\mathcal{D}_n$ that belong to B, and $\bar{\phi}_B^{\mathcal{D}_n,w} = 1/n^{\mathcal{D}_n,w}(B) \sum_{i\in\mathcal{D}_n} W_i\phi(Y_i)\mathbb{1}_{X_i\in B}$ the weighted mean of $\phi(Y_i)$ for the observations within B (we set $\bar{\phi}_B^{\mathcal{D}_n,w} = +\infty$ if $n^{\mathcal{D}_n,w}(B) = 0$).

Let $A \subset \mathcal{X}$ be a node of a tree, $j \in \{1, \dots, p\}$ a variable, and u real number. Define:

$$L^w(u, j, A, \mathcal{D}_n) = \frac{1}{n^{\mathcal{D}_n,w}(A)} \cdot \sum_{i\in\mathcal{D}_n} W_i \cdot \left(\phi(Y_i) - \bar{\phi}_{A_l}^{\mathcal{D}_n,w}\mathbb{1}_{X_i^{(j)}\le u} - \bar{\phi}_{A_r}^{\mathcal{D}_n,w}\mathbb{1}_{X_i^{(j)}>u}\right)^2 \cdot \mathbb{1}_{X_i\in A},$$

where $A_l = A \cap \{X^{(j)} \le u\}$ and $A_r = A \cap \{X^{(j)} > u\}$.

We also define $n^{\mathcal{D}_n,\hat{w}}(B)$, $\bar{\phi}_B^{\mathcal{D}_n,\hat{w}}$ and $L^{\hat{w}}(u, j, A, \mathcal{D}_n)$ as the same quantities with the weights W_i replaced by the weights $\hat{W}_i$.

The binary split chosen at node A is $A = A_l^* \cup A_r^*$, with $A_l^* = A \cap \{X^{(j^*)} \leq u^*\}$, $A_r^* = A \cap \{X^{(j^*)} > u^*\}$, and (u^*, j^*) given by:

$$(u^*, j^*) = \underset{\substack{j \in \{1,..,p\} \\ u \in \left\{X_i^{(j)} \,:\, i \in \mathcal{D}_n \text{ s.t. } X_i \in A\right\}}}{\operatorname{argmin}} L^{\hat{w}}(u, j, A, \mathcal{D}_n).$$

For a heuristic understanding of the algorithm, consider a fixed set A. We abusively assume that A is independent from the data, which is not correct in practice since A is selected from the data. Under this assumption, we can study the asymptotics of $L^w(u, j, A, \mathcal{D}_n)$ since the observations $(Y_i, X_i)_{i \in \mathcal{D}_n}$ which belong to A are i.i.d. and follow the conditional distribution $\mathcal{L}((Y, X)|X \in A)$. Until the end of this section, we assume that A is independent from the data.

With A_z standing for either A_l or A_r, we have from the law of large numbers and equation (1),

$$\bar{\phi}_{A_z}^{\mathcal{D}_n,w} \underset{n \to +\infty}{\longrightarrow} E\left[W \cdot \phi(Y)|X \in A_z\right] = E\left[\phi(T)|X \in A_z\right],$$

and

$$\begin{aligned}\frac{1}{n^{\mathcal{D}_n,w}(A_z)} \cdot \sum_{\substack{i \in \mathcal{D}_n \\ X_i \in A_z}} W_i \cdot \left(\phi(Y_i) - \bar{\phi}_{A_z}^{\mathcal{D}_n,w}\right)^2 &\underset{n \to +\infty}{\longrightarrow} E\left[W \cdot \left(\phi(Y) - E[W \cdot \phi(Y)|X \in A_z]\right)^2 \middle| X \in A_z\right] \\ &= E\left[\left(\phi(T) - E[\phi(T)|X \in A_z]\right)^2 \middle| X \in A_z\right].\end{aligned}$$

Then, by splitting the sum in $L^w(u, j, A, \mathcal{D}_n)$ between A_l and A_r, we get:

$$L^w(u, j, A, \mathcal{D}_n) \underset{n \to +\infty}{\longrightarrow} L^\infty(u, j, A),$$

with

$$\begin{aligned}L^\infty(u, j, A) \;=\; &\frac{P(X \in A_l)}{P(X \in A)} \cdot E\left[(\phi(T) - E[\phi(T)|X \in A_l])^2 \middle| X \in A_l\right] \;+ \\ &\frac{P(X \in A_r)}{P(X \in A)} \cdot E\left[(\phi(T) - E\left[\phi(T)|X \in A_r\right])^2 \middle| X \in A_r\right].\end{aligned}$$

Therefore we observe that, at the cost of replacing $\hat{W}_i$ by W_i, the selected split is asymptotically the one which minimizes the sum of the within variances of the two child nodes, weighted by their relative importance.

It is also important to see that given $\mathcal{D}_n$ and the splitting criteria L, the growing of a tree is deterministic. Therefore we can define a function $\mathcal{T}(\mathcal{D}, L)$ which outputs a tree given a sample and a splitting criterion.

2.3.2 Tree prediction

The growing of a binary tree $\mathcal{T}$ results in a partition $(\mathcal{X}_k)_{k=1,..,K}$ of the input space. In order to predict the mean value of the target variable, a natural estimator is $m^{\hat{w}}(x) = \sum_{k=1}^{K} \bar{\phi}_{\mathcal{X}_k}^{\mathcal{D}_n,\hat{w}} \mathbb{1}_{x\in\mathcal{X}_k}$. Under **H1**/**H2** we have for the denoised version m^w,

$$m^w(x) \underset{n\to+\infty}{\longrightarrow} \sum_{k=1}^{K} E\left[\phi(T)|X \in \mathcal{X}_k\right] \mathbb{1}_{x\in\mathcal{X}_k},$$

which shows that $m^{\hat{w}}$ approximates a piecewise constant estimator of f.

However, though the growing step is done using a least-square criterion – which is well-adapted to mean regression – the prediction made in the terminal leaf may not necessarily be a sample mean (e.g., it may be a quantile – see Meinshausen (2006)). We can formalize this by defining a function $\mathcal{M}(\mathcal{T}, \mathcal{D}, m)$ which outputs a predictor given a binary tree, a sample, and a type of terminal leaf-based estimator. In the following section, we propose a second terminal leaf estimator for the mean within a leaf.

Both for tree prediction and split selection, note that the method we propose is a generalization of the classical CART regression algorithm. Indeed, setting the weights W_i all equal to 1 results in Breiman's original algorithm. We note L and m the split criteria and the terminal leaf estimator obtained when setting the weights W_i all equal to 1.

2.3.3 Description of the algorithms

The tools we developed in previous sections can be used in different random forest algorithms. We have studied three survival weighted random forests (*swRF*) in this article: *swRF1*, *swRF2* and *swRF3*. These algorithms differ in the way weights $\hat{W}_i$ are introduced.

In *swRF1* the weights $\hat{W}_i$ are taken into account in the bootstrap part of the random forest.

Based on the whole training set, we build $\hat{S}_C$ (or $\hat{S}_C(\cdot|X)$) and deduce $(\hat{W}_i)_{i=1,\dots,n}$. Then, all bootstrap samples of the forest are drawn with replacement and with sampling probabilities proportional to the weights $(\hat{W}_i)_{i=1,\dots,n}$. A CART tree is then grown on each bootstrap sample, using the classical algorithm (i.e. with L and m which denote the splitting criterion and the terminal leaf estimator with uniform weights).

In algorithms *swRF2* and *swRF3*, bootstrap samples are drawn uniformly with replacement, as it is done in the classical random forest algorithm, and the weights are computed on the bootstrap samples. This means that the weights of a given observation may vary from tree to tree. The only difference between *swRF2* and *swRF3* is the terminal leaf estimator used. In *swRF2*, $m^{\hat{w}}$ is used as described in Section 2.3.2, whereas in *swRF3* we use $m^{\hat{w}_{loc}}(x) = \sum_{k=1}^{K} \bar{\phi}_{\mathcal{X}_k}^{\mathcal{D}_n,\hat{w}_{loc}} \mathbb{1}_{x\in\mathcal{X}_k}$. In fact, $m^{\hat{w}_{loc}}$ is similar to $m^{\hat{w}}$ except that the weights used in each terminal leaf are computed using a Kaplan-Meier estimator inside the leaf.

Pseudocode of the three algorithms we study is shown in Algorithms 1 and 2 below.

Algorithm 1: *swRF*1

Input : Data : $(Y_i, \delta_i, X_i)_{i=1,\dots,n}$, $M > 0$: number of trees

Output: Ensemble predictor *swRF*1

Compute weights $(\hat{W}_i)_{i=1,\dots,n}$: $\hat{W}_i = \delta_i / \hat{S}_C(Y_i)$ (or $\hat{W}_i = \delta_i / \hat{S}_C(Y_i|X_i)$)

for $j = 1, \dots, M$ **do**

  Draw $\mathcal{D}_{n,j}$: sample n observations w.r.t. weights $(\hat{W}_i)_{i=1,\dots,n}$, with replacement

  Build $\mathcal{T}_j = \mathcal{T}(\mathcal{D}_{n,j}, L)$

  $\hat{m}_j = \mathcal{M}(\mathcal{T}_j, \mathcal{D}_{n,j}, m)$

end

return : $\hat{m}_{swRF1} = \frac{1}{M} \sum_{i=1}^{M} \hat{m}_j$

Algorithm 2: *swRF*2 & *swRF*3

Input : Data : $(Y_i, \delta_i, X_i)_{i=1,\dots,n}$, $M > 0$: number of trees, $mode \in \{1, 2\}$: type of terminal leaf estimator

Output: Ensemble predictor *swRF*2 (or *swRF*3)

for $j = 1, \dots, M$ **do**

  Draw $\mathcal{D}_{n,j}$: sample n observations uniformly with replacement

  Compute weights $(\hat{W}_i^{\mathcal{D}_{n,j}} : i \in \mathcal{D}_{n,j})$: $\hat{W}_i^{\mathcal{D}_{n,j}} = \delta_i / \hat{S}_C^{\mathcal{D}_{n,j}}(Y_i)$ $\left(\text{or } \hat{W}_i^{\mathcal{D}_{n,j}} = \delta_i / \hat{S}_C^{\mathcal{D}_{n,j}}(Y_i|X_i)\right)$

  Build $\mathcal{T}_j = \mathcal{T}(\mathcal{D}_{n,j}, L^{\hat{w}})$

  $\hat{m}_j = \mathcal{M}(\mathcal{T}_j, \mathcal{D}_{n,j}, m^{\hat{w}})$ if $mode = 1$ ($\hat{m}_j = \mathcal{M}(\mathcal{T}_j, \mathcal{D}_{n,j}, m^{\hat{w}_{loc}})$ if $mode = 2$)

end

return : $\hat{m}_{swRF(2,3)} = \frac{1}{M} \sum_{i=1}^{M} \hat{m}_j$

Remark 1 *For each of swRF*1*, swRF*2 *or swRF*3*, we can build three models which correspond to the three ways to estimate the weights* $(W_i)_{i=1,\dots,n}$ *developed in Section 2.2.2.*

A practical limitation of the weighted random forest algorithm is that for a large non-censored observation Y_i*, the corresponding weight tends to be very big (Cui, Zhu, & Kosorok (2017)). This is even more so in the conditional case if* X *is multidimensional, since the estimated survival function* $\hat{S}_C(\cdot|X = x)$ *may decrease quickly for some values of* x*. Overly-large weights may give too much importance to a single observation, both in terms of split selection and terminal leaf estimation.*

We propose two ways to deal with this problem. The first is to threshold the weights $\hat{W}_i$ *so that the maximum ratio between two nonzero weights does not exceed some value. Let* r_{max} *be the maximum ratio allowed between two nonzero IPCW weights. The choice of the value for* r_{max} *corresponds to a typical bias-variance trade-off. Indeed, thresholding the weights introduces bias in the swRF procedure since large non-censored observations get smaller weights, and thus predicted values tend to be lower, but on the other hand this lowers the estimator's variance, which is high when some observations have very large weights.*

The second strategy is to use swRF to model the time $T' = min(T, T_{max})$ *and not the original time* T*. This is natural because we are working on the estimation of the expected value for* T*, and the problem of estimating the average of a right-censored time is generally hard to solve; indeed, there is high variance in the estimation of the tail of the distribution, which causes fluctuations in the estimated mean.*

In the following, we will combine the two strategies to achieve good accuracy using swRF.

2.3.4 Parameters of the random forest algorithm

In our earlier presentation of random forests, we made no mention of the parameters which need to be specified to build them. Since there is a large number of parameters, and since parameters may differ according to the random forest implementation in question, we only mention the most common ones – those which have been shown to have the strongest impact on the resulting model.

The main parameters in a random forest are summarized in Tab. 1. The parameters *minleaf* and *maxdepth* impact tree size in the forest, and we discuss their influence on the performances of the algorithms in supplementary material. As for the parameter *mtry*, it is especially important in high dimensional settings.

Parameter	Description
$minleaf$	minimum value for the number of observations that should be present in a leaf; a split that would result in a leaf of less than $minleaf$ observations cannot be selected
$maxdepth$	length of the longest downward path from the root node to a leaf of the tree; a tree only consisting of the root node has depth 0.
$mtry$	number of candidate variables that are randomly sampled at each node when choosing the best split

Tab. 1: Parameters of the random forest algorithm.

2.4 Assessing the quality of a model's fit

In the applications which follow, our strategy to evaluate a fitted model relies on train-test approaches: let $\mathcal{D}_{tr}$ and $\mathcal{D}_{te}$ be train and test sets of indices. However, due to the right censoring, the usual accuracy criterion used in regression settings cannot be computed on the test set.

In fact, since the problems of bias on the test and train samples are similar, we can use for model validation the same sample fitting method with IPCW that we use for model training. Let $(\hat{W}_i)_{i\in\mathcal{D}_{te}}$ be the estimated IPCW and $\hat{f}$ the predictor function of a fitted model. Then, the quantity $1/n_{te} \cdot \sum_{i\in\mathcal{D}_{te}} \hat{W}_i \cdot (\phi(Y_i) - \hat{f}(X_i))^2$ is an estimator of the mean squared error (MSE) of the model $\hat{f}$ (with $n_{te} = Card(\mathcal{D}_{te})$ the number of test observations). This is the method suggested in Gerds & Schumacher (2006) and the one we adopt in this article. One disadvantage is that again this approach raises the question of the choice of weights to consider for IPCW; in the same way as for training, we choose here to compute the validation error corresponding to three types of weight: KM, Cox and RSF. Therefore, we use not one but three criteria to compare model performance. Computation of the IPCW weights is performed separately on the training and test sets.

The problem of model validation in the presence of censoring has received considerable attention in the literature, and other performance measures exists. The C-index (Harrell et al. (1982)), which generalizes the Kendall tau for right-censored data and is related to the area under the ROC curve (Heagerty & Zheng (2005)), is very popular, so we also compute it when comparing models.

3 Simulated data example

3.1 Technical details

3.1.1 Data simulation

In our simulations, the covariates X are distributed as marginal uniform distributions on $[-1, 1]$ linked through a Gaussian copula with covariance given by an $AR(1)$ covariance matrix with coefficient ρ, i.e., a matrix K^ρ such that $K^\rho_{i,j} = \rho^{|i-j|}$. The coefficient ρ is random and uniformly distributed on $[0, 0.6]$.

We investigate three experimental settings which correspond to three different models for $\mathcal{L}(T|X)$, the distribution of T given X. In the following description of the three cases, λ_T and k_T are constants that will be specified later, and $Weibull(\lambda, k)$ refers to the distribution with density $g(u) = k/\lambda \cdot (u/\lambda)^{k-1} e^{-(u/\lambda)^k} \mathbb{1}_{u\geq 0}$:

- Case 1 (Weibull): $T|(X, \beta_T) \sim Weibull(\lambda_T e^{-{}^t\beta_T \cdot X}, k_T)$, with $\beta_T \sim N(\mu = 0, \sigma^2 = 0.04 \cdot I_p)$ and $\beta_T \perp\!\!\!\perp X$.

- Case 2 (Independent mixture of Weibulls): $T|(X, G, (\beta_{T,j})_{j=1,...,4}) \sim Weibull(\lambda_T e^{-{}^t\beta_{T,G} \cdot X}, k_T)$ with:

  - $G \sim Unif\{1, 2, 3, 4\}$ and $G \perp\!\!\!\perp X, (\beta_{T,j})_{j=1,...,4}$,
  - $(\beta_{T,j})_{j=1,...,4}$ i.i.d. with $\forall j \in [\![1, 4]\!]$, $\beta_{T,j} \sim N(\mu = 0, \sigma^2 = 0.09 \cdot I_p)$ and $(\beta_{T,j})_{j=1,...,4} \perp\!\!\!\perp X$.

- Case 3 (Covariate-dependent mixture of Weibulls, $p \geq 2$):

  $$T|(X, G, (\beta_{T,j})_{j=1,...,4}) \sim Weibull(\lambda_T e^{-{}^t\beta_{T,G} \cdot X}, k_T)$$

  with:

  - $G = \mathbb{1}_{X_1\geq 0, X_2\geq 0} + 2 \cdot \mathbb{1}_{X_1\geq 0, X_2<0} + 3 \cdot \mathbb{1}_{X_1<0, X_2\geq 0} + 4 \cdot \mathbb{1}_{X_1<0, X_2<0}$,
  - $(\beta_{T,j})_{j=1,...,4}$ i.i.d. with $\forall j \in [\![1, 4]\!]$, $\beta_{T,j} \sim N(\mu = 0, \sigma^2 = 0.04 \cdot I_p)$ and $(\beta_{T,j})_{j=1,...,4} \perp\!\!\!\perp X$.

The nature of $\mathcal{L}(C|X)$ is the same in every setting: $C|X \sim Weibull(\lambda_C e^{-{}^t\beta_C \cdot X}, k_C)$ with λ_C and β_C fitted to satisfy certain conditions (see Algorithm 3).

In each setting, we test the algorithm for two different functions ϕ: $\phi(t) = t$ and $\phi(t) = log(t+1)$. We also assess the sensitivity of the results to the censoring rate of the simulated data ($q = 0.1, 0.3$ or 0.5) and to the strength of the dependence between C and X (measured in terms of percentage of explained variance: $R2_C = 0.05$ or 0.1).

The process of data simulation is detailed in Algorithm 3.

Algorithm 3: Simulated data generation

Input : n : number of simulated observations
p : dimension of X
q : proportion of censored observations
$R2_C$: proportion of explained variance for $C|X$
k_C : shape parameter of C
(λ_T, k_T) : scale and shape parameter for T
T_{max} : threshold for T

Output: Dataset of generated data

1 Draw ρ uniformly on $[0, 0.6]$
2 Simulate $(X_i)_{i\in1,\dots,n}$ i.i.d. with $X \sim GaussianCopula(K^\rho)$ and marginals $Unif[-1, 1]$
3 Generate $(T_i)_{i=1,\dots,n}$ from case 1, 2 or 3
4 Draw $\beta_{C_0} \sim N(\mu = 0, \sigma^2 = 0.04 \cdot I_p)$, let $\beta_C = \eta \cdot \beta_{C_0}$, and calibrate η so that the empirical estimate of the explained variance $\widehat{R2}_C$ satisfies $\widehat{R2}_C \approx R2_C$
5 Calibrate λ_C so that the empirical proportion of censored observations $\hat{q}$ satisfies $\hat{q} \approx q$
6 Generate $(C_i)_{i=1,\dots,n}$ with $\forall i,\ C_i \sim Weibull(\lambda_C e^{-{}^t\beta_C \cdot X_i}, k_C)$
7 Check on the simulated $(C_i)_{i=1,\dots,n}$ that $\widehat{R2}_C \approx R2_C$ and $\hat{q} \approx q$
8 Build $\forall i = 1, \dots, n$, $Y_i = min(T_i, C_i)$, $\delta_i = \mathbb{1}_{T_i \leq C_i}$
9 Build $\forall i = 1, \dots, n$, $T'_i = min(T_i, T_{max})$, $Y'_i = min(T'_i, C_i)$, $\delta'_i = \mathbb{1}_{T'_i \leq C_i}$ (cf. Section 2.2.2. Remark: $\hat{q}$ at step 5 is derived from T' and not T)

return : Dataset with columns X, T, C, Y, δ, T', Y', δ'

We take $n = 2000$, $p = 6$, $\lambda_T = 100$, $k_T = 1$, and $k_C = 0.8$ as parameters for the data simulation. The parameter T_{max} is set to 1000.

3.1.2 Different models

In our experiments, we compare the performance of sixteen models. First, we have three *swRF1* models, associated with Kaplan-Meier (*swRF11*), Cox (*swRF12*) and random survival forest (*swRF13*) weights, respectively.

For *swRF2* and *swRF3*, many weight computations are necessary since weights are computed on the bootstrap samples. The use of RSF to compute the weights is thus computationally intensive, so for these two models we only use KM (*swRF21* and *swRF31*) and Cox (*swRF22* and *swRF32*) weights.

We also test five other algorithms to compare with *swRF*. They all follow the same idea: first, fit a model to estimate $S_T(\cdot|X)$ (which gives an estimator $\hat{S}_T(\cdot|X)$), and second, integrate $\hat{S}_T(\cdot|X)$ to get an estimator of $E[\phi(T)|X = x]$: $\hat{f}(x) = -\int \phi \cdot d\hat{S}_T(\cdot|X = x)$. Five different models are used to estimate $S_T(\cdot|X)$ at the first step : the Cox model, the RSF algorithm, the forest of relative risk trees (RRT) algorithm of Ishwaran et al. (2004), and two versions of the reinforcement learning trees (RLT) algorithm described in the Section 4 of Zhu (2013) (one which uses reinforcement learning and one which does not). This gives five benchmarks for *swRF* that we call respectively *RSFr*, *Cr*, *RRTr*, *RLTr* (with reinforcement) and *nRLTr* (without reinforcement).

Since here we are working with simulated data, it is interesting to consider, as a baseline indicator, the performance of the random forest algorithm as if there was no censoring in the data. This random forest is trained on the non-censored data $(X_i, T_i)_{i=1,..n}$, and the associated model is denoted *RF*. Similarly, it is interesting to consider the exact IPCW weights W_i, as described in Section 2.2.1, for the use of *swRF*. The exact IPCW weights correspond to theoretical weights computed using the true conditional survival function of the censoring $S_C(\cdot|X)$ which is known in our simulated data application. This gives three additional comparison models called *swRF14*, *swRF24* and *swRF34*.

So that the different random forest models involved are comparable, all of the random forests in the simulations are set with the same values for the three parameters we evoked in Section 2.3.4. The parameter *mtry* is set to 6 so that $mtry = p$ and there is no randomness involved in split selection. We also take $maxdepth = 4$ and $minleaf = 50$, as justified in supplementary material.

The *swRF* models are trained on the threshold data (Y', δ', X) with $Y' = \min(T', C)$, setting $r_{max} = 50$ (see the definitions of T' and r_{max} in the Remark 1 of Section 2.3.3). *RF* is trained on the data (T', X), whereas Cr, $RSFr$, $RRTr$, $RLTr$ and $nRLTr$ are trained on the data (Y, δ, X). For the latter models, we then compute the prediction at the point x with $\hat{f}(x) = -\int \phi \cdot d\hat{S}_{T'}(\cdot|X = x)$, where $\hat{S}_{T'}(t|X) = \hat{S}_T(t|X) \cdot \mathbb{1}_{t \leq T_{max}}$.

3.2 Results and analysis

3.2.1 Performance of the models

To compare the models, we generated 100 simulated datasets of 2000 observations with Algorithm 3. For each iteration, we train all models on the same 1000 observations, and evaluate them on the 1000 remaining ones. Model accuracy is measured in terms of the mean squared

error (MSE), which can can be computed in the simulated data context since we have access to all of the $\phi(T_i)$ values. The means of the MSE over the 100 i.i.d. replicates of the simulation process are given in Fig. 1. To keep the figure visually understandable, we chose to represent in the main text only the results for the most illuminating models. Nevertheless, we emphasize that the following analysis is consistent with the complete results, obtained for the sixteen models, that are given in supplementary material.

The results on simulated data help us to learn more about the various models. An initial observation is that the results are quite affected by the ϕ function being considered, and less by changes in the distribution of $\mathcal{L}(T|X)$. Of course, RF is generally the best model, except when the constrained form of the Cox model is well-suited to the problem, whereby Cr may outperform RF. Recall that the RF algorithm should only be seen as a benchmark that cannot be used in practice, since it relies on the complete (uncensored) observations. For $q = 0.1$, the performance of the $swRF$ models is very close to that of RF, which is natural since $swRF$ is equivalent to RF in the non-censored case (i.e., $q = 0$).

We can also see that $swRF$ models are more sensitive to an increase in censoring rate than the comparison methods which model directly $S_T(\cdot|X)$, namely: Cr, $RSFr$, $RRTr$ and $RLTr$. The $swRF$ models perform well for $q = 0.3$ and 0.1, but the MSE is much larger when $q = 0.5$. For $\phi(t) = t$, $RSFr$ and Cr perform very well overall, but for $\phi(t) = log(t+1)$, the $swRF$ models are usually more accurate, especially when $q = 0.1$ and $q = 0.3$, or the data is simulated as a covariate-dependent mixture (Case 3).

We can also compare the $swRF$ models with each other. We first remark that the $swRF$ scores are organized in terms of the groups of $swRF$ ($swRF1$ and $swRF3$). While for $\phi(t) = t$ and $q = 0.5$ $swRF3$ models achieve high MSE due to particular iterations where $swRF3$ does not work well, the results for $\phi(t) = log(t + 1)$ are very different for $swRF1$ and $swRF3$. The difference is mostly due to the terminal leaf estimator used (this is confirmed by the results of $swRF2$ in supplementary material). Indeed, we can see that the results of $swRF3$ are closer to the results of $RSFr$, $RRTr$ and $RLTr$, and the terminal leaf estimators used in these models are almost the same; $swRF3$ relies on a within leaf nonparametric estimation of the distribution of T using a KM estimator (i.e. $m^{\hat{W}_{loc}}$), whereas the other models rely on a Nelson-Allen estimator (Aalen (1978)) as explained in Ishwaran & Kogalur (2007). Also, the type of weights we consider has a second-order impact on the results, which becomes more significant with larger censoring rates. However, it is interesting to note that comparing the results of $swRF11$ and $swRF13$, we see that conditional weights computed with RSF tend to give better results than the Kaplan-Meier ones. By comparing the results of $swRF32$ and $swRF34$, we observe that

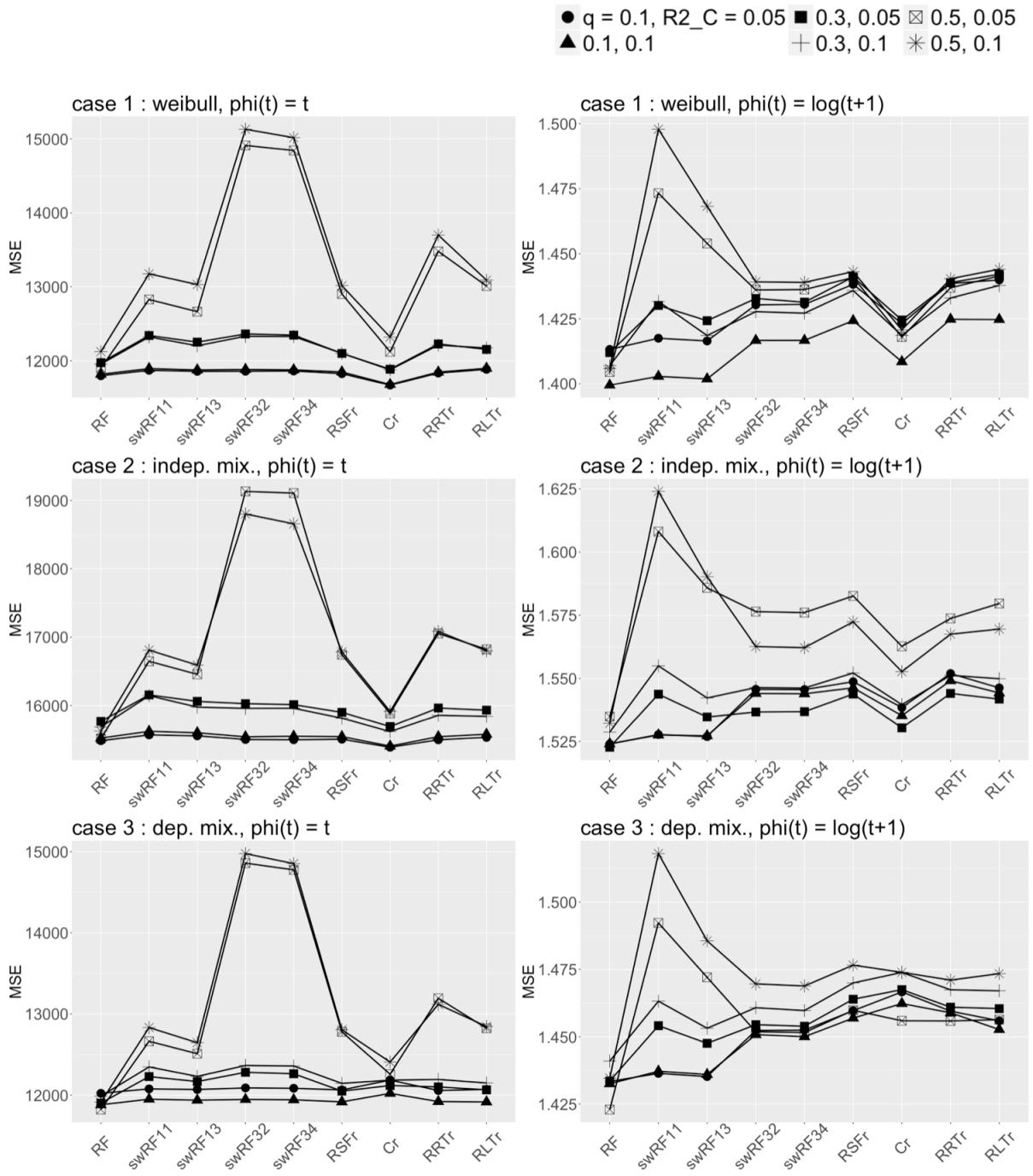

Fig. 1: Results on simulated data.

For each setting, the mean of the MSE values over 100 i.i.d. replicates of the simulation process is shown. The censoring rate q is equal to 0.1, 0.3, or 0.5, while the percentage of explained variance of C given X: $R2_C$, is set to 0.05 or 0.1.

Cox IPCW and exact IPCW lead to very close MSE. This shows that Cox IPCW (but also RSF IPCW as we will see in the next section) are good proxies for exact IPCW.

3.2.2 Correlations between weighted MSE (IPCW) and MSE

The results shown in Fig. 1 are estimated MSE which do not suffer from the effects of censoring. Since such estimators are not available for real data studies, in practice we rely – as explained in Section 2.4 – on weighted estimators (involving KM, Cox, RSF, or uniform weights) when comparing models. Therefore, it is of interest to compute, in the simulated data case, correlations between the non-censored MSE and weighted approximations of it. We added to the results the correlations with the exact (theoretical) weights which are available in the simulated data case. The average values of these correlations are displayed in Tab. 2. These results show that RSF and Cox weights better replicate the non-censored MSE than the KM ones, and therefore offer better comparison criteria for model selection. Let us note that RSF and Cox weights are even more accurate than exact weights. Such type of phenomena have already been observed in other censored regression models. For example, in the case of linear regression, Koul et al. (1981) noticed that the asymptotic variance of their slope estimator was smaller using KM weights rather than the true distribution function of the censoring (see their Remark 4.5 p. 1280). As a possible interpretation, as pointed by Koul et al. (1981), one may claim that the information contained in the censored observations would be completely lost if we were using exact weights, while estimation of the distribution of the censoring relies on the censored observations. On the other hand, the criterion based on uniform weighting of the non-censored observations has little correlation with the non-censored criterion, so we should not rely on it.

q	$R2_C$	RSF weights	Cox weights	exact weights	KM weights	unif. weights
0.10	0.05	0.91	0.91	0.90	0.90	0.70
0.10	0.10	0.92	0.92	0.91	0.91	0.70
0.30	0.05	0.78	0.77	0.72	0.74	0.20
0.30	0.10	0.76	0.75	0.70	0.66	0.18
0.50	0.05	0.72	0.71	0.66	0.65	0.14
0.50	0.10	0.61	0.64	0.59	0.49	0.09

Tab. 2: Correlations between weighted MSE (IPCW) and MSE.

Mean Spearman correlations between the non-censored and weighted (IPCW) MSE, as a function of q and $R2_C$. For each of the six cases, we have 600 calculations of the MSE and weighted MSE (2 choices for $\phi \times 3$ choices for $\mathcal{L}(T|X) \times 100$ iterations), and we calculate correlations between different fitting criteria.

4 Real data application

4.1 Modeling the churn behavior of policy holders

The problem of predictive modeling has gained interest in recent years in the insurance community, and optimization of underwriting is one relevant application (Frees et al. (2014)). For insurance contract brokers, forecasting customer value represents an opportunity to improve margins in a competitive environment (Cummins & Doherty (2006), Maas (2010)). Moreover, churn modeling is important for the estimation of customer value, as described in Verhoef & Donkers (2001). Here, we focus on applying the methodology developed in Section 2 to build a predictive model of the impact of churn on prospect value (as defined in Section 1) as predicted by insurance brokers.

A broker's approach to estimate the prospect values is given by the following multiplicative formula, where the hats indicate we are referring to estimated quantities:

$$\widehat{value} = \widehat{p}_{sub} \cdot \widehat{pr} \cdot \widehat{f}_{ew} \cdot \widehat{f}_{ch},$$

with $\widehat{p}_{sub}$ the probability of subscription, $\widehat{pr}$ the predicted premium, $\widehat{f}_{ew}$ the early withdrawal factor, and $\widehat{f}_{ch}$ the churn factor. We are only interested in modeling f_{ch} here. In this factorization, the churn factor only depends on the termination time of the contract via a function ϕ_{ch} as shown in Fig. 2.

The data that we study here is the customer base of an insurance broker from 1^{st} October 2009 to 31^{st} July 2016, and we focus in particular on complementary health insurance contracts. Data are available about the effective dates of the contracts, current states of contracts (active, terminated), and termination dates of contracts (if terminated). Before the underwriting, prospective information is given as 6 variables: age (8 levels starting from 18 years old), gender (2 levels: female, male), number of children insured (5 levels: 0, 1, 2, 3, $\geq$ 4), social security regime (7 levels: agricultural employee, employee, retired, retired self-employed, self-employed, student, unemployed), level of insurance (3 levels: low, medium, high), geographical zone (4 levels: Ile-de-France region, north, southern, other). Other client characteristics are available in the database but are only known after subscription occurs, hence it is impossible to use them to evaluate prospects.

We use this information to predict churn factors, which correspond to the variable $\phi(T)$. This variable is censored for a contract which is still active on January 15^{th}, 2016. The censoring

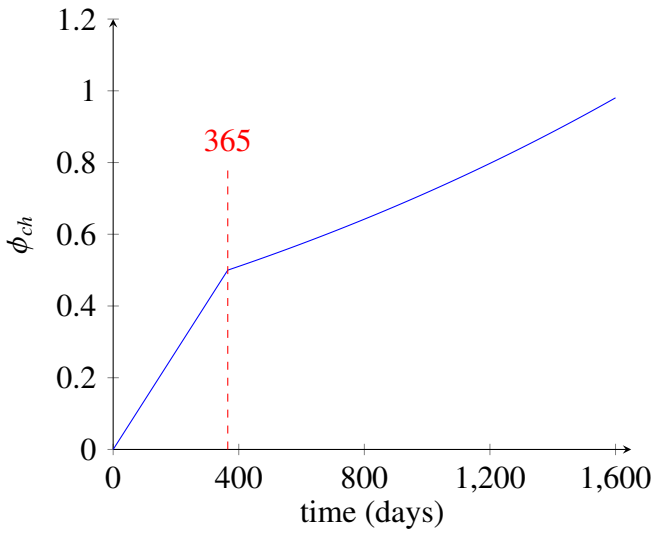

Fig. 2: The churn factor as a function of termination time.

Here, ϕ_{ch} is the broker's commission, expressed per unit of annual premium. It divides into two parts: a withholding part during the first year – which reaches 50% of the annual premium after one year – and a subsequent part, starting after one year, equal to 10% of annual premiums (taking into account annual resets of the premium).

variable C is the age of the contract, i.e., the duration between the date of effect of the contract and January 15^{th}, 2016. The 6 variables above constitute the covariate vector X.

4.2 Additional details about the experiments

The same models as in the simulated data application (see Section 3.1.2) are compared in this real data application, excluding RF and the $swRF$ models based on the exact weights ($swRF14$, $swRF24$, and $swRF34$), which cannot be used in the real data case. Also, T_{max} is set at 1465 days; this makes sense from the point of view of this application since we are then estimating the expected return of a prospect within the four years following subscription. All $swRF$, $RSFr$, $RRTr$ and $RLTr$ models are set with the parameters $mtry = 6$, $maxdepth = 5$, and $minleaf = 100$. For $swRF$ models, r_{max} is set to 50.

We test the various models with four different ϕ functions: $\phi = \phi_{ch}$ (Fig. 2), $\phi(t) = log(t+1)$, $\phi(t) = t$, and $\phi(t) = \mathbb{1}_{t>380}$. For the latter function, the choice of the threshold 380 is related to the willingness to consider churns that occur in the first contract year. Most of these churns occur before 365 days, but a significant number is reported a few days later, explaining the margin of 15 days that we take (T_{max} was set to 381 in this case since, from 381 days on information on $\phi(T)$ is not censored).

4.3 Results and analysis

As in the simulated data case, we use a train-test approach (5000 observations each) for model testing, with 100 repetitions. Each training and testing set are drawn uniformly without replacement from the original dataset in such a way that the training and test sets do not overlap. The results in terms of MSE computed with KM, Cox and RSF weights, and on the C-index, are shown in Fig. 3. We give the results for the same models as in Section 3.2.1 except for *RF* and *swRF*34 which are not usable anymore with real data, and for *swRF*22 that we choose to incorporate into the figure. The results for the other models are given in supplementary material.

We first observe that the results for $\phi = \phi_{ch}$, $\phi(t) = log(t+1)$, and $\phi(t) = t$ look very similar, especially for $\phi = \phi_{ch}$ and $\phi(t) = log(t+1)$. Hence, the strong influence of the function ϕ that we noticed for the simulated data does not carry over to the real data. Moreover, there is a tendency in the results that *swRF* models fitted using a certain type of weights perform very well with the MSE computed using the same type of weights. This is especially clear for the Cox and KM weights (*swRF*11 and *swRF*22 models), indicating that the type of weights considered has a real impact on the estimated distribution of (T, X).

We choose to consider the RSF weighted estimation of the MSE as the reference criteria for comparing the models. Indeed, we saw in Section 3.2.2 that Cox and RSF weights give approximations of the MSE that are the most correlated with the MSE computed on non-censored data. In addition, there is a risk of overfitting the type of weights we use, and since RSF weights are only involved in the model *swRF*13, they constitute a good choice. In Fig. 4 and Tab. 3, we show the performance of each model, calculated with RSF weights and for $\phi = \phi_{ch}$. Fig. 4 indicates the model *swRF*32 achieves on average the lowest error, followed by $RLTr$, an ordering confirmed by the table of rank statistics.

The C-index's statistic (Harrell et al. (1982)) corresponds to the proportion of ordered pairs in the test set which are well-ordered by the model $\hat{f}$, i.e. :

$$\frac{Card\left(\{(i,j) \in \mathcal{D}_{te}^2 : \hat{f}(X_i) > \hat{f}(X_j), Y_i > Y_j, \delta_j = 1\}\right)}{Card\left(\{(i,j) \in \mathcal{D}_{te}^2 : Y_i > Y_j, \delta_j = 1\}\right)}.$$

We can see in Fig. 3 that $RSFr$, Cr, $RRTr$ and $RLTr$ achieve the best C-index out of the set of models. This illustrates that it is important to consider a quadratic error criterion rather than a rank one when the goal is to estimate a mean value. It is the case in our situation since we are interested in getting the best prediction for the churn factor $f_{ch} = \phi_{ch}(T')$. The C-index values nevertheless show the benefit of using conditional weights within the *swRF* algorithm in order to get models that rank the observations well.

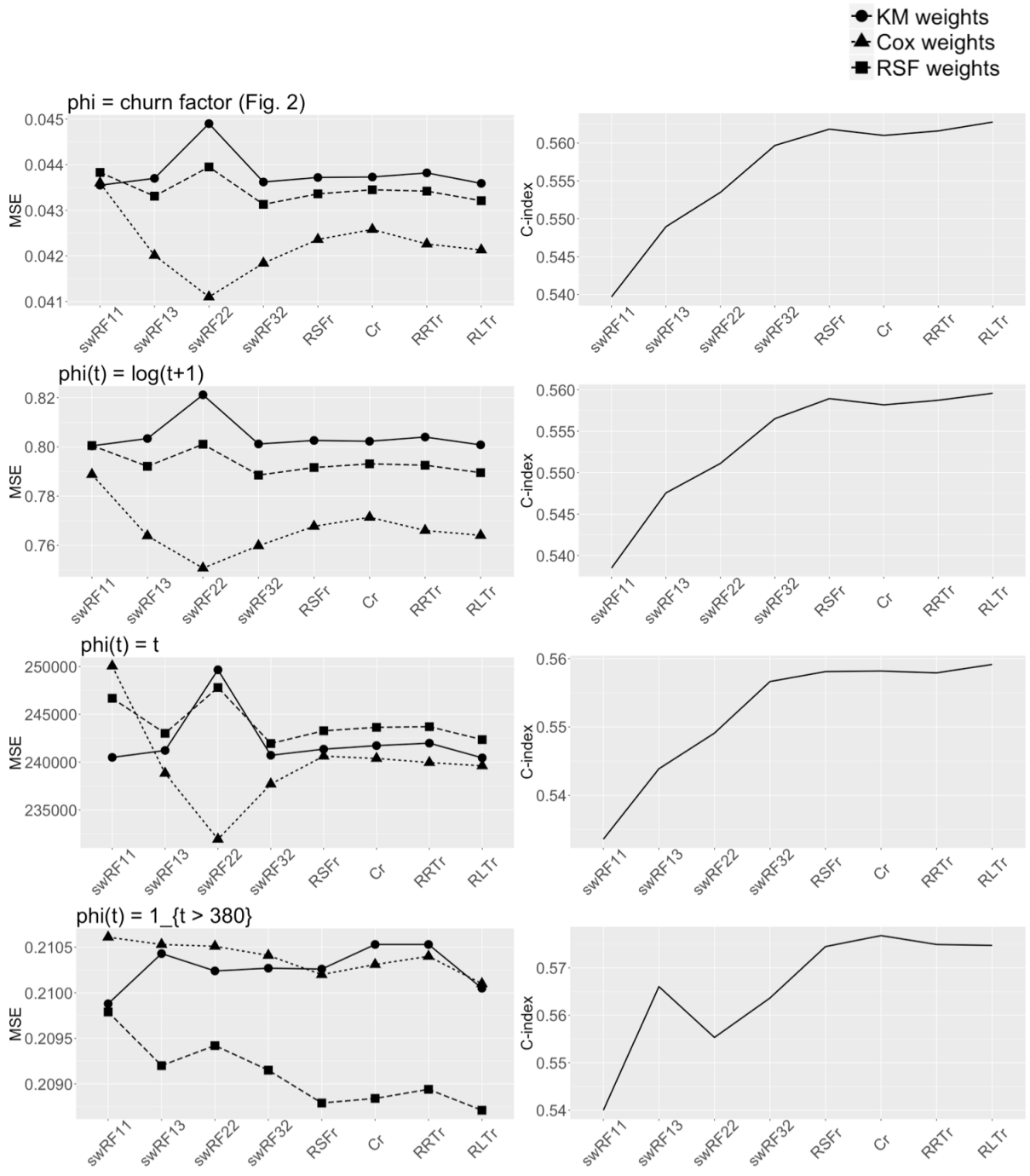

Fig. 3: Results on real data.

Left : the mean of the MSE values over 100 i.i.d. replicates of the simulation process, computed with KM, Cox or RSF weights. Right : the mean of the C-index over 100 i.i.d. replicates.

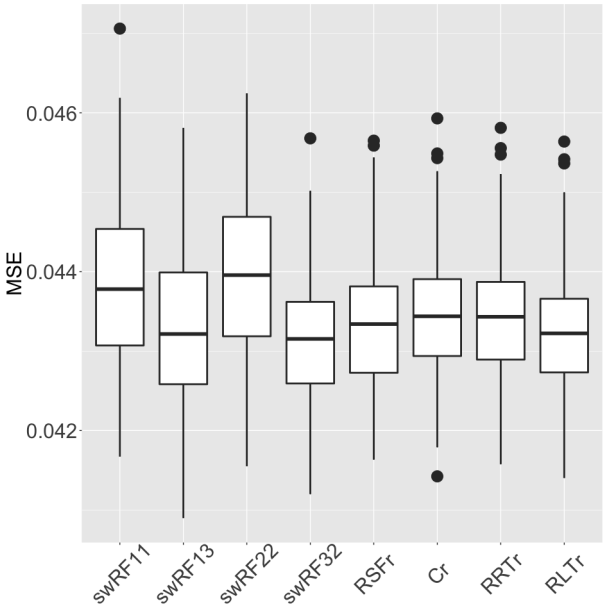

Fig. 4: Boxplot of the performances (MSE) of the models; $\phi = \phi_{ch}$, with RSF weights.

model	mean rank
$swRF11$	9.0
$swRF13$	4.8
$swRF22$	9.6
$swRF32$	2.8
$RSFr$	5.9
Cr	6.9
$RRTr$	6.9
$RLTr$	3.7

Tab. 3: Mean ranks of the models; $\phi = \phi_{ch}$, with RSF weights.

5 Conclusion

In this paper, we have considered a class of weighted random forest algorithms, where the weight put on each observation is designed to compensate for censoring. A classical issue in censored regression models is the identifiability assumption that defines the dependence structure between the censoring and the variables involved in the model. Therefore, we have distinguished between two situations, namely the case where censoring is independent, and the case where this variable is allowed to depend on the duration variable T through the covariates. The latter case is the more general of the two, but leads to difficulties in computing appropriate weights, due to the strong impact of the dimension of the covariate vector. For this reason, we have proposed a compromise by modeling the conditional censoring distribution using a Cox model or a RSF model. We showed on simulated data that our method is competitive with other statistical methods in terms of accuracy. Also, this approach appeared to give the most accurate results in our application to modeling the commission received by an insurance broker, which is a non linear function of the time at which an insurance policyholder surrenders their contract. Moreover, it gives more accurate results than competing approaches such as random survival forest, relative risk forest, and reinforcement learning trees, in the setting we have considered. Finally, the weighting strategies we have proposed easily generalize to other regression-based approaches such as, to give one example, quantile regression.

As future work, it would be interesting to investigate the performances of the weighted random forest method in a setting where the dimension of the covariate vector is higher, which is an important application case (Zhu (2013), Ishwaran et al. (2010)). Indeed, in this situation the estimation of the conditional IPCW is hard and may require to adapt the algorithm. A suitable method might be to sample, for each tree of the forest, a subset of covariates to take into account in the computation of the conditional IPCW. Also, doubly robust survival trees studied in Steingrimsson et al. (2016) seems to be a great research direction. Promising results are already presented in Steingrimsson et al. (2018) using a relative risk tree to estimate $S_C(\cdot|X)$, and we might wonder if we could observe the same phenomenon in the results as we obtain in our work regarding the influence of the estimation of $S_C(\cdot|X)$: i.e. an improvement when using a conditional estimator instead of the KM estimator, and a benefit of using ensemble models such as RSF. Finally, theoretical work may be of interest as well; since the weighted random forest generalizes the (non-censored) random forest for regression, it may be possible to transpose consistency results obtained in the non-censored case (e.g. in Scornet et al. (2015)) to the weighted case.

Software

All analyses were performed with the R package `sword` (`github.com/YohannLeFaou/sword`) which was developed for the purposes of this article. The code used for producing the results is available at the address: `github.com/YohannLeFaou/impact-churn-health-insurance`.

Acknowledgements

We would like to thank the company Santiane, who provided the data that served for the experiments, and was always available to answer our questions.Also, we greatly thank the two referees for their valuable comments and their insightful suggestions.

References

Aalen, O. (1978). Nonparametric inference for a family of counting processes. *The Annals of Statistics*, 701–726.

Ahn, H., & Loh, W.-Y. (1994). Tree-structured proportional hazards regression modeling. *Biometrics*, 471–485.

Beran, R. (1981). *Nonparametric regression with randomly censored survival data* (Tech. Rep.). Univ. California, Berkeley.

Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test*, *25*(2), 197–227.

Bou-Hamad, I., Larocque, D., & Ben-Ameur, H. (2011). A review of survival trees. *Statistics Surveys*, *5*, 44–71.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Ciampi, A., Thiffault, J., Nakache, J.-P., & Asselain, B. (1986). Stratification by stepwise regression, correspondence analysis and recursive partition: a comparison of three methods of analysis for survival data with covariates. *Computational statistics & data analysis*, *4*(3), 185–204.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society*, *34*, 187–220.

Cui, Y., Zhu, R., & Kosorok, M. (2017). Tree based weighted learning for estimating individualized treatment rules with censored data. *Electronic journal of statistics*, *11*(2), 3927–3953.

Cui, Y., Zhu, R., Zhou, M., & Kosorok, M. (2017). Consistency of survival tree and forest models: splitting bias and correction. *arXiv preprint arXiv:1707.09631v4*.

Cummins, J. D., & Doherty, N. A. (2006). The economics of insurance intermediaries. *Journal of Risk and Insurance*, *73*(3), 359–396.

Dabrowska, D. M. (1989, 09). Uniform consistency of the kernel conditional kaplan-meier estimate. *Ann. Statist.*, *17*(3), 1157–1167. Retrieved from `https://doi.org/10.1214/aos/1176347261`

Davis, R. B., & Anderson, J. R. (1989). Exponential survival trees. *Statistics in Medicine*, *8*(8), 947–961.

Frees, E. W., Derrig, R. A., & Meyers, G. (2014). *Predictive modeling applications in actuarial science* (Vol. 1). Cambridge University Press.

Fu, W., & Simonoff, J. S. (2016). Survival trees for left-truncated and right-censored data, with application to time-varying covariate data. *Biostatistics*, *18*(2), 352–369.

Gerds, T. A., & Schumacher, M. (2006). Consistent estimation of the expected brier score in general survival models with right-censored event times. *Biometrical Journal*, *48*(6), 1029–1040.

Goldberg, Y., & Kosorok, M. R. (2017). Support vector regression for right censored data. *Electronic Journal of Statistics*, *11*(1), 532–569.

Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L., & Rosati, R. A. (1982). Evaluating the yield of medical tests. *Jama*, *247*(18), 2543–2546.

Heagerty, P. J., & Zheng, Y. (2005). Survival model predictive accuracy and roc curves. *Biometrics*, *61*(1), 92–105.

Hothorn, T., Bühlmann, P., Dudoit, S., Molinaro, A., & Van Der Laan, M. J. (2006). Survival ensembles. *Biostatistics*, *7*(3), 355–373.

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics*, *15*(3), 651–674.

Hothorn, T., Lausen, B., Benner, A., & Radespiel-Tröger, M. (2004). Bagging survival trees. *Statistics in medicine*, *23*(1), 77–91.

Ishwaran, H., Blackstone, E. H., Pothier, C. E., & Lauer, M. S. (2004). Relative risk forests for exercise heart rate recovery as a predictor of mortality. *Journal of the American Statistical Association*, *99*(467), 591–600.

Ishwaran, H., & Kogalur, U. B. (2007). Random survival forests for r. *R news*, *7*.

Ishwaran, H., & Kogalur, U. B. (2010). Consistency of random survival forests. *Statistics & probability letters*, *80*(13-14), 1056–1064.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008). Random survival forests. *The annals of applied statistics*, 841–860.

Ishwaran, H., Kogalur, U. B., Gorodeski, E. Z., Minn, A. J., & Lauer, M. S. (2010). High-dimensional variable selection for survival data. *Journal of the American Statistical Association*, *105*(489), 205–217.

Kaplan, E. L., & Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, *53*(282), 457–481.

Koul, H., Susarla, V. v., & Van Ryzin, J. (1981). Regression analysis with randomly right-censored data. *The Annals of Statistics*, 1276–1288.

Lagakos, S. (1979). General right censoring and its impact on the analysis of survival data. *Biometrics*, 139–156.

LeBlanc, M., & Crowley, J. (1992). Relative risk trees for censored survival data. *Biometrics*, 411–425.

LeBlanc, M., & Crowley, J. (1993). Survival trees by goodness of split. *Journal of the American Statistical Association*, *88*(422), 457–467.

Li, A. H., & Bradic, J. (2019). Censored quantile regression forests. *arXiv preprint arXiv:1902.03327*.

Lopez, O., Milhaud, X., & Thérond, P.-E. (2016). Tree-based censored regression with applications in insurance. *Electronic journal of statistics*, *10*(2), 2685–2716.

Lopez, O., Patilea, V., & Van Keilegom, I. (2013). Single index regression models in the presence of censoring depending on the covariates. *Bernoulli*, *19*(3), 721–747.

Maas, P. (2010). How insurance brokers create value—a functional approach. *Risk Management and Insurance Review*, *13*(1), 1–20.

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

Molinaro, A. M., Dudoit, S., & Van der Laan, M. J. (2004). Tree-based multivariate regression and density estimation with right-censored data. *Journal of Multivariate Analysis*, *90*(1), 154–177.

Scornet, E., Biau, G., & Vert, J.-P. (2015). Consistency of random forests. *The Annals of Statistics*, *43*(4), 1716–1741.

Segal, M. R. (1988). Regression trees for censored data. *Biometrics*, 35–47.

Steingrimsson, J. A., Diao, L., Molinaro, A. M., & Strawderman, R. L. (2016). Doubly robust survival trees. *Statistics in medicine*, *35*(20), 3595–3612.

Steingrimsson, J. A., Diao, L., & Strawderman, R. L. (2018). Censoring unbiased regression trees and ensembles. *Journal of the American Statistical Association*, 1–14.

Stute, W. (2003). Kaplan–meier integrals. In *Advances in survival analysis* (Vol. 23, p. 87 - 104). Elsevier. Retrieved from http://www.sciencedirect.com/science/article/pii/S0169716103230054

Van der Laan, M. J., & Robins, J. M. (2003). *Unified methods for censored longitudinal data and causality*. Springer Science & Business Media.

Verhoef, P. C., & Donkers, B. (2001). Predicting customer potential value an application in the insurance industry. *Decision support systems*, *32*(2), 189–199.

Zhu, R. (2013). *Tree-based methods for survival analysis and high-dimensional data* (Unpublished doctoral dissertation). The University of North Carolina at Chapel Hill.

Zhu, R., & Kosorok, M. R. (2012). Recursively imputed survival trees. *Journal of the American Statistical Association*, *107*(497), 331–340.

Zhu, R., Zeng, D., & Kosorok, M. R. (2015). Reinforcement learning trees. *Journal of the American Statistical Association*, *110*(512), 1770–1784.