npcure: An R Package for Nonparametric Inference in Mixture Cure Models

by Ana López-Cheda, M. Amalia Jácome, Ignacio López-de-Ullibarri

Abstract Mixture cure models have been widely used to analyze survival data with a cure fraction. They assume that a subgroup of the individuals under study will never experience the event (cured subjects). So, the goal is twofold: to study both the cure probability and the failure time of the uncured individuals through a proper survival function (latency). The R package **npcure** implements a completely nonparametric approach for estimating these functions in mixture cure models, considering right-censored survival times. Nonparametric estimators for the cure probability and the latency as functions of a covariate are provided. Bootstrap bandwidth selectors for the estimators are included. The package also implements a nonparametric covariate significance test for the cure probability, which can be applied with a continuous, discrete, or qualitative covariate.

1 Introduction

In classical survival analysis, it is assumed that all the individuals will eventually experience the event of interest. However, there are many contexts in which this assumption might not be true. Noticeable examples are the lifetime of cancer patients after treatment, time to infection in a risk population, or time to default in credit scoring, among many others. Cure models are a stream of methods recently developed in survival analysis that take into account the possibility that subjects could never experience the event of interest. See Maller and Zhou (1996) for early references and Amico and Van Keilegom (2018) for an updated review.

Let $\mathbf{X}$ be a set of covariates and Y the time to the event of interest with conditional survival function $S(t|\mathbf{x}) = P(Y > t|\mathbf{X} = \mathbf{x})$. Mixture cure models, initially proposed by Boag (1949), consider the population as a mixture of two types of subjects: the susceptible of experiencing the event if followed for long enough ($Y < \infty$) and the cured ones ($Y = \infty$). Hence, the survival function of Y can be written as

$$S(t|\mathbf{x}) = 1 - p(\mathbf{x}) + p(\mathbf{x}) S_0(t|\mathbf{x}),$$

where $1 - p(\mathbf{x}) = P(Y = \infty|\mathbf{X} = \mathbf{x}) = \lim_{t\to\infty} S(t|\mathbf{x})$ is the cure probability, and the (proper) survival function of the uncured subjects or *latency* is $S_0(t|\mathbf{x}) = P(Y > t|Y < \infty, \mathbf{X} = \mathbf{x})$. A major advantage of these models over the non-mixture approach is that they allow the covariates to have different effect on cured and uncured individuals.

The cure probability, $1 - p(\mathbf{x})$, is usually estimated parametrically by assuming a logistic form $\log(p(\mathbf{x}) / (1 - p(\mathbf{x}))) = \boldsymbol{\beta}'\mathbf{x}$, with $\boldsymbol{\beta}$ a parameter vector. Estimation of $S_0(t|\mathbf{x})$ can be done by assuming a particular parametric distribution for the failure time of the uncured subjects, or more generally, by applying, e.g., proportional hazards (PH) or accelerated failure time (AFT) assumptions. These two approaches lead to parametric (see, e.g., Farewell, 1982, 1986; Denham et al., 1996) or semiparametric (see, e.g., Kuk and Chen, 1992; Peng et al., 1998; Peng and Dear, 2000; Li and Taylor, 2002) mixture cure models.

An attractive feature of parametric and semiparametric models is that they provide close expressions for relevant parameters and functions. On the other hand, the sound inference is guaranteed only if the chosen model fits the data suitably. A problem with these methods is that the parametric assumptions may be incorrect. For example, regarding the cure rate $1 - p(\mathbf{x})$, there is no reason to believe that the cure rate is monotone in $\mathbf{x}$, let alone that it follows a logistic model. To solve this hassle, Müller and Van Keilegom (2019) propose a test statistic to assess whether the cure rate, as a function of $\mathbf{X}$, satisfies a certain parametric model. As for the latency function, $S_0(t|\mathbf{x})$, it is difficult to verify the distributional assumptions of the model. The goodness of fit for the latency function has only been addressed in settings without covariates and in an informal way (Maller and Zhou, 1996). The challenge of developing procedures for testing the parametric form of the conditional survival function of the uncured with covariates is even more ambitious. It would lead to curse-of-dimensionality problems and remains an open question.

As a result of the increasing demand for the use of cure models, the number of packages in R accounting for the possibility of cure in survival analysis has grown significantly over the last decade: see the CRAN task view on survival analysis (`https://CRAN.R-project.org/view=Survival`). The **smcure** package (Cai et al., 2012) fits the semiparametric PH and AFT mixture cure models (see Kalbfleisch and Prentice, 2002). Besides, the **NPHMC** package (Cai et al., 2013) allows to calculate the sample size of a survival trial with or without cure fractions. More recently, the **flexsurvcure** package (Amdahl, 2017) provides flexible parametric mixture and non-mixture cure models for time-

to-event data, and the **rcure** package (Han et al., 2017) incorporates methods related to robust cure models for survival data which include a weakly informative prior in the logistic part. The **geecure** package (Niu and Peng, 2018) features the marginal parametric and semiparametric PH mixture cure models for analyzing clustered survival data with a possible cure fraction. Furthermore, the **miCoPTCM** package (Bertrand et al., 2020) fits semiparametric promotion time cure models with possibly mis-measured covariates, while the **mixcure** package (Peng, 2020) implements parametric and semiparametric mixture cure models based on existing R packages. For interval-censored data with a cure fraction, the **GORcure** package (Zhou et al., 2017) implements the generalized odds rate mixture cure model, including the PH mixture cure model and the proportional odds mixture cure model as special cases. The **intercure** package (Brettas, 2016) provides an implementation of semiparametric cure rate estimators for interval-censored data using bounded cumulative hazard and frailty models.

In contrast with (semi)parametric procedures, nonparametric methods do not rely on data belonging to any particular parametric family or fulfilling any parametric assumption. They estimate the goal functions without making any assumptions about its shape, so they have much wider applicability than alternative parametric methods. A completely nonparametric mixture cure model must consider purely nonparametric estimators for both the cure rate, $1 - p(\mathbf{x})$, and latency function, $S_0(t|\mathbf{x})$. Unlike the (semi)parametric approach, nonparametric mixture cure models have been under study only in recent years. Laska and Meisner (1992), building on the Kaplan-Meier (KM) product-limit (PL) estimator of the survival function $S(t) = P(Y > t)$ (Kaplan and Meier, 1958), derive nonparametric estimators of the cure rate and latency function, but their model does not allow for covariates. More recently, Xu and Peng (2014) propose a nonparametric estimator of the cure rate with one or more covariates, showing its consistency and asymptotic normality. This estimator was further studied by López-Cheda et al. (2017a), who, besides proving that it is the maximum likelihood nonparametric estimator of the cure probability, also obtain an i.i.d. representation and proposed a bootstrap-based bandwidth selector. As for the latency function, López-Cheda et al. (2017b) introduce a completely nonparametric estimator, studied some theoretical properties, and proposed a bandwidth selector based on the bootstrap.

Although some of the aforementioned packages have a nonparametric flavor, their approach to mixture cure modeling is not completely nonparametric. Our R package **npcure** (López-de-Ullibarri et al., 2020) fills the gap by providing implementations of the nonparametric estimator of the cure rate function proposed by Xu and Peng (2014) (further studied by López-Cheda et al., 2017a) and of the nonparametric estimator of the latency function proposed by López-Cheda et al. (2017b).

Furthermore, the generalized PL estimator of the conditional survival function, $S(t|x)$, proposed by Beran (1981), is implemented. Note that the estimators of the cure rate and latency implemented in **npcure** relate strongly to Beran estimator. In any case, Beran estimator is of outstanding importance by its own, as evidenced by the variety of R packages with functions for computing it, like, e.g., `Beran()` in package **condSURV** (Meira-Machado and Sestelo, 2016), `prodlim()` in package **prodlim** (Gerds, 2018) and `Beran()` in package **survidm** (Meira-Machado et al., 2019). The function in our package compares advantageously with the aforementioned functions with respect to the issue of bandwidth selection. This smoothing parameter plays an essential role in the bias-variance tradeoff of every nonparametric smoothing method. In Dabrowska (1992), an expression for the bandwidth minimizing the asymptotic mean squared error (MSE) of this estimator was obtained, and a plug-in bandwidth selector was proposed based on suitable estimators of the unknown functions in that expression. However, the performance of this bandwidth selector is unsatisfying for small sample sizes, and a cross-validation (CV) procedure is usually preferred (see Iglesias-Pérez, 2009; Gannoun et al., 2007, among others). Recently, Geerdens et al. (2017) propose an improved CV bandwidth selector, especially with a high censoring rate. To the best of our knowledge, there are not any R packages allowing to compute Beran estimator with a suitable bandwidth selector: while the **condSURV** and **survidm** packages do not consider any bandwidth selectors, the **prodlim** package uses nearest neighborhoods as the smoothing parameter. The **npcure** package, available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=npcure`, fulfills this need with the implementation of the CV bandwidth selector for the Beran estimator in Geerdens et al. (2017).

In this paper, we explain how the **npcure** package can be used in the context of nonparametric mixture cure models with right-censored data. The main objective is to estimate the cure probability and latency functions, as well as to perform covariate significance tests for the cure rate. In the next section, we describe our approach to nonparametric estimation in mixture cure models. The methodology applied in the covariate significance tests is presented in another section. Two sections follow, devoted respectively to explain the package functions and to illustrate their use with an application to a medical dataset.

2 Nonparametric estimation in mixture cure models

One of the specificities of time-to-event data is related to the presence of individuals that have not experienced the event by the end of the study. The observed survival times of these individuals are said to be *right-censored* and underestimate the true unknown time to the occurrence of the event. This situation is usually modeled by considering a censoring variable C, with distribution function G, which is conditionally independent of Y given the covariate $\mathbf{X}$. The observed data are then $\{(\mathbf{X}_i, T_i, \delta_i) : i = 1, \ldots, n\}$, where $T = \min(Y, C)$ is the observed lifetime and $\delta = \mathbf{1}(Y \leq C)$ is the uncensoring indicator. For a one-dimensional continuous covariate X, Xu and Peng (2014) propose the following nonparametric kernel-type estimator of the cure rate:

$$1 - \hat{p}_h(x) = \prod_{i=1}^{n} \left(1 - \frac{\delta_{[i]} B_{h[i]}(x)}{\sum_{r=i}^{n} B_{h[r]}(x)}\right) = \hat{S}_h(T^1_{\max}|x), \tag{1}$$

where, for $i = 1, \ldots, n$, $\delta_{[i]}$ and $X_{[i]}$ are the concomitant status indicator and covariate corresponding to the ith ordered time $T_{(i)}$, and

$$B_{h[i]}(x) = \frac{K_h\left(x - X_{[i]}\right)}{\sum_{j=1}^{n} K_h\left(x - X_{[j]}\right)} \tag{2}$$

are the Nadaraya-Watson weights, where $K_h(\cdot) = \frac{1}{h}K\left(\frac{\cdot}{h}\right)$ is a rescaled kernel with bandwidth $h \to 0$. Although some different kernel functions could be considered, the Epanechnikov kernel, defined as

$$K(u) = \frac{3}{4}(1 - u^2)\mathbf{1}(|u| \leq 1),$$

is the one implemented in the **npcure** package. Moreover, $\hat{S}_h$ is the estimator of the conditional survival function S in Beran (1981), and $T^1_{\max} = \max_{\{i:\delta_i=1\}} T_i$ is the largest uncensored failure time. Xu and Peng (2014) prove the consistency and asymptotic normality of the estimator in (1), and López-Cheda et al. (2017a) show that it is the local maximum likelihood estimator of the cure rate, and obtained an i.i.d. representation and an asymptotic expression for the MSE.

The nonparametric latency estimator proposed by López-Cheda et al. (2017a), and further studied in López-Cheda et al. (2017b), is:

$$\hat{S}_{0,b}(t|x) = \frac{\hat{S}_b(t|x) - (1 - \hat{p}_b(x))}{\hat{p}_b(x)}, \tag{3}$$

where $\hat{S}_b$ is the PL estimator of the conditional survival function S (Beran, 1981) and $\hat{p}_b$ is the estimator in (1). As in the case of the cure rate estimator, a smoothing parameter b, not necessarily equal to h, is needed to compute $\hat{S}_{0,b}$ in (3).

Consistency of the nonparametric estimators

The proposed nonparametric estimators of both the cure rate and latency are consistent under the general condition (see Laska and Meisner, 1992; Maller and Zhou, 1992; López-Cheda et al., 2017a,b)

$$\tau_0 \leq \tau_G(x), \tag{4}$$

where $\tau_0 = \sup_x \tau_0(x)$, and $\tau_0(x) = \sup\{t \geq 0 : S_0(t|x) > 0\}$ and $\tau_G(x) = \sup\{t \geq 0 : G(t|x) < 1\}$ are the right endpoints of the support of the conditional distribution of the uncures and the censoring variable, respectively.

The condition in (4) ensures $1 - p(x)$ and $S_0(t|x)$ to be consistently estimated when there is zero probability that a susceptible individual survives beyond the largest possible censoring time, $\tau_G(x)$. Since $T^1_{\max}$ converges to τ_0 in probability (see Xu and Peng, 2014), assumption (4) guarantees that, asymptotically, all times observed after the largest uncensored survival time, $T^1_{\max}$, can be assumed to correspond to cures.

Under condition (4), $S_0(\tau_G(x)|x) = 0$ and, for large n, the cure rate estimator in (1) tends to a nonparametric estimator of $S(\tau_G(x)|x) = 1 - p(x) + p(x)S_0(\tau_G(x)|x) = 1 - p(x)$. However, if there could be uncured individuals surviving beyond $\tau_G(x)$, then $S_0(\tau_G(x)|x) > 0$ and the estimator in (1) would estimate $S(\tau_G(x)|x) = 1 - p(x) + p(x)S_0(\tau_G(x)|x) > 1 - p(x)$. This might happen, for example, in a clinical trial with fixed maximum follow-up time.

These comments emphasize that care must be exercised in choosing the length of follow-up if

cures might be present since too much censoring or insufficient follow-up time could lead to erroneous conclusions. For example, if the last observation is uncensored, then, even if there is considerable late censoring, the estimated cure rate is 0. To avoid these difficulties, particularly with heavy censoring, reasonably long follow-up times and large sample sizes may be required. In this way, $S_0\left(\tau_G(x)|x\right)$ is sufficiently small for the cure rate estimator in (1) to be close enough to $1-p(x)$.

Thus, when estimating $1-p(x)$ and $S\left(t|x\right)$ for a given x with a data set, it is important to be confident that $\tau_0 \leq \tau_G(x)$. In any case, if the censoring distribution $G(t|x)$ has a heavier tail than $S_0\left(t|x\right)$, the cure rate estimates computed with the nonparametric estimator in (1) will tend to have smaller biases regardless of the value of $\tau_0(x)$ (see Xu and Peng, 2014). Maller and Zhou (1992) propose a simple nonparametric test to assess condition (4). The procedure is based on the length of the interval $(T^1_{\max}, T_{(n)}]$, i.e., the right tail of the KM estimate where it has a constant value. A long plateau with heavy censoring at the right tail of the KM curve is interpreted as evidence that follow-up time has been long enough to conclude that condition (4) holds.

Bandwidth selection

The nonparametric estimators in (1) and (3) depend on two smoothing parameters, h and b, respectively. Bootstrap-based selectors for the bandwidth h of the cure rate estimator and the bandwidth b of the latency estimator are proposed by López-Cheda et al. (2017a) and López-Cheda et al. (2017b), respectively. The bandwidths are locally chosen so that the selected bandwidths h_x and b_x depend on the point x of estimation. Using locally adaptive bandwidths instead of global ones is advantageous because they adapt to the structure of the underlying function, differentially smoothing its flat and peaky parts.

For a fixed value x, the bootstrap bandwidth of the cure estimator, h^*_x, was introduced by López-Cheda et al. (2017a) as the minimizer of the bootstrap MSE, approximated with B resamples as follows:

$$MSE^*_x(h_x) \simeq \frac{1}{B}\sum_{b=1}^{B}\left(\hat{p}^{*b}_{h_x}(x) - \hat{p}_{g_x}(x)\right)^2, \tag{5}$$

where $\hat{p}^{*b}_{h_x}(x)$ is the estimator of $p(x)$ in (1) computed with $\left\{\left(X^{*b}_i, T^{*b}_i, \delta^{*b}_i\right) : i = 1,\ldots,n\right\}$ (the bth bootstrap resample), and using the local bandwidth h_x, and $\hat{p}_{g_x}(x)$ is computed with the original sample $\{(X_i, T_i, \delta_i) : i = 1,\ldots,n\}$, and the local pilot bandwidth g_x.

With respect to the latency estimator in (3), López-Cheda et al. (2017b) propose to choose the bandwidth b_x locally with a bootstrap bandwidth selector. The bootstrap bandwidth of the latency estimator, b^*_x, is taken as the minimizer of the bootstrap mean integrated squared error (MISE):

$$MISE^*_x(b_x) \simeq \frac{1}{B}\sum_{b=1}^{B}\int_0^u \left(\hat{S}^{*b}_{0,b_x}\left(t|x\right) - \hat{S}_{0,g_x}\left(t|x\right)\right)^2 dt, \tag{6}$$

where $\hat{S}^{*b}_{0,b_x}\left(t|x\right)$ is the nonparametric estimator of $S_0\left(t|x\right)$ in (3) computed with the bth bootstrap resample and local bandwidth b_x, $\hat{S}_{0,g_x}\left(t|x\right)$ is the same estimator obtained using the original sample and a local pilot bandwidth g_x, and u is an adequately chosen upper bound of the integral.

For a fixed covariate value x, the procedure for obtaining the bootstrap bandwidth selector of h_x for $\hat{p}_{h_x}(x)$ (respectively, b_x for $\hat{S}_{0,b_x}\left(t|x\right)$) is as follows:

1. Generate B bootstrap resamples $\left\{\left(X^{*b}_i, T^{*b}_i, \delta^{*b}_i\right) : i = 1,\ldots,n\right\}$, for $b = 1,\ldots,B$.
2. Consider a search grid of bandwidths $h_l \in \{h_1,\ldots,h_L\}$. For $b = 1,\ldots,B$ and $l = 1,\ldots,L$, compute the nonparametric estimator $\hat{p}^{*b}_{h_l}(x)$ (respectively, the nonparametric latency estimator, $\hat{S}^{*b}_{0,h_l}\left(t|x\right)$) with the bth bootstrap resample and bandwidth h_l.
3. Compute the nonparametric estimator $\hat{p}_{g_x}(x)$ (respectively, the nonparametric latency estimator $\hat{S}_{0,g_x}\left(t|x\right)$) with the original sample and pilot bandwidth g_x.
4. For each bandwidth $h_l \in \{h_1,\ldots,h_L\}$, compute the Monte Carlo approximation of $MSE^*_x(h_l)$ in (5), (respectively, the Monte Carlo approximation of $MISE^*_x(h_l)$ in (6)).
5. The bootstrap bandwidth h^*_x for the cure rate estimator (respectively, b^*_x for the latency estimator) is the minimizer of the Monte Carlo approximation of $MSE^*_x(h_l)$ (respectively, $MISE^*_x(h_l)$) over the grid of bandwidths $\{h_1,\ldots,h_L\}$.

Following López-Cheda et al. (2017a) and López-Cheda et al. (2017b), the bootstrap resamples in Step 1 are generated considering the following procedure, which is equivalent to the simple weighted bootstrap proposed by Li and Datta (2001) without resampling the covariate X:

I. Generate $X_1^*, \ldots, X_n^*$ by fixing $X_i^* = X_i$, $i = 1, \ldots, n$.

II. For each i, compute the weighted empirical distribution $\hat{F}_{g_{X_i^*}}\left(t, \delta | X_i^*\right)$ with the original sample, where $\hat{F}_{g_x}\left(t, \delta | x\right) = \sum_{i=1}^{n} B_{g_x i}(x) \mathbf{1}\left(T_i \leq t, \delta_i \leq \delta\right)$ and $B_{g_x i}(x)$ is computed with a local pilot bandwidth g_x (see (7) below).

III. For each i, generate the pair $\left(T_i^*, \delta_i^*\right)$ from the weighted empirical estimator $\hat{F}_{g_{X_i^*}}\left(t, \delta | X_i^*\right)$ of the conditional distribution.

López-Cheda et al. (2017a) and López-Cheda et al. (2017b) show that the effect of the pilot bandwidth on the bootstrap bandwidth selectors of h_x and b_x is considerably low. Consequently, the same expression for the pilot bandwidth, g_x, is used in Step II of the bootstrap resampling procedure and in the approximation of the MSE_x^* in (5) for the selection of the bandwidth h_x of the cure rate estimator (respectively, in the approximation of the $MISE_x^*$ in (6) for the bandwidth b_x of the latency estimator):

$$g_x = \frac{d_k^+(x) + d_k^-(x)}{2} 100^{1/9} n^{-1/9}, \tag{7}$$

where $d_k^+(x)$ (respectively, $d_k^-(x)$) is the distance from x to the kth nearest neighbor on the right (respectively, on the left). If there are not at least k neighbors on the right (or left), we use $d_k^+(x) = d_k^-(x)$. López-Cheda et al. (2017a) show that a good choice for the parameter k is to consider $k = n/4$. The order $n^{-1/9}$ satisfies the conditions in Theorem 1 of Li and Datta (2001) and coincides with the optimal order for the pilot bandwidth obtained by Cao and González-Manteiga (1993) in the case without censoring.

When selecting locally adaptive bandwidths, the results might look a little bit spiky due to its local nature (see, e.g., Brockmann et al., 1993, on local bandwidth selection for kernel regression estimators). That could be the case for the bootstrap bandwidths for both the cure rate and latency functions. To get rid of the fluctuation of these local bandwidths, h_x and b_x can be further smoothed, for example, by computing a centered moving average of the unsmoothed vector of bandwidths as in López-Cheda et al. (2017a).

3 Covariate significance tests

In medical studies, it is usually important to assess whether the cure probability depends on a specific covariate, X. Noting that the cure rate can be interpreted as the regression function $E\left(\nu | X = x\right) = 1 - p(x)$, where ν is the indicator of cure, the question can be cast in the form of a hypothesis test:

$$\begin{cases} H_0 : E\left(\nu | X\right) = 1 - p \\ H_1 : E\left(\nu | X\right) = 1 - p(X) \end{cases}. \tag{8}$$

Although there are some parametric approaches to deal with this hypothesis testing problem (see Müller and Van Keilegom, 2019, among others), the only completely nonparametric method was introduced by López-Cheda et al. (2020). Their procedure is based on the test for selecting explanatory variables in nonparametric regression described by Delgado and González-Manteiga (2001). The greatest advantage of the proposed significance test for the cure rate is that although the test is completely nonparametric, no smoothing parameters are required to test (8).

The main challenge when testing (8) is that the cure indicator, ν, is only partially known due to censoring: complete observations are known to be uncured ($\nu = 0$), but censored observations might be either cured or uncured (i.e., ν is unknown). Under right censoring, all of the cured individuals and some of the uncured ones will be censored. This makes it difficult to guess whether a censored observation belongs to the cured or uncured subpopulation. López-Cheda et al. (2020) solved this situation by replacing the unknown and inestimable response variable ν in (8) by an unknown but estimable response η with the same conditional expectation as ν:

$$\eta = \frac{\nu\left(1 - \mathbf{1}(\delta = 0, T \leq \tau)\right)}{1 - G\left(\tau | X\right)}, \tag{9}$$

where τ is an unknown time beyond which a lifetime might be assumed to be cured. López-Cheda et al. (2020) propose to estimate η by replacing G and τ with suitable nonparametric estimators. The censoring distribution is estimated with the generalized PL estimator by Beran (1981) computed with the cross-validation (CV) bandwidth selector in Geerdens et al. (2017) when X is continuous and with the stratified KM estimator with the same bandwidth selector otherwise. The cure threshold, τ, is estimated as $\hat{\tau} = T_{\max}^1$, the largest uncensored observed time. The expression of η in (9) avoids the need for an estimator of the unknown cure indicator, ν, since if $\delta_i = 1$ or $(\delta_i = 0, T_i < \hat{\tau})$ then $\hat{\eta}_i = 0$,

whereas if $(\delta_i = 0, T_i \geq \hat{\tau})$ then $\hat{\eta}_i = 1/\left(1 - \hat{G}(\hat{\tau}|X_i)\right)$. It is easy to check that $E(\nu|X) = E(\eta|X)$ if the conditional censoring distribution $G(t|x)$ is independent of the cure status.

Finally, building on Delgado and González-Manteiga (2001) and using the estimated values of η in (9), the significance test proposed by López-Cheda et al. (2020) is based on the process:

$$U_n(x) = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\eta}_i - \frac{1}{n}\sum_{j=1}^{n}\hat{\eta}_j\right)\mathbf{1}(X_i \leq x). \tag{10}$$

Cramér-von Mises (CM) or Kolmogorov-Smirnov (KS) test statistics can be used:

$$\begin{aligned} CM_n &= \sum_{i=1}^{n} U_n^2(X_i), \\ KS_n &= \max_{i=1,\ldots,n} n^{1/2}|U_n(X_i)|. \end{aligned} \tag{11}$$

Note that if X is a nominal variable, it is impossible to compute the indicator function in (10). In this case, López-Cheda et al. (2020) propose to consider all the possible 'ordered' permutations of the values of X and to compute $U_n(x)$ according to the 'ordering' of each permutation. The values of the CM and KS test statistics are given by the maximum of the values CM_n and KS_n computed along with all the permutations.

The distribution of the CM and KS statistics under the null hypothesis is approximated by bootstrap, according to the following steps:

A. Obtain X_i^*, $i = 1, \ldots, n$, by randomly resampling with replacement from $\{X_1, \ldots, X_n\}$.

B. Estimate the probability of cure under H_0 as $1 - \hat{p} = \hat{S}_n^{KM}\left(T_{\max}^1\right)$, with $\hat{S}_n^{KM}$ the KM estimator of the survival function $S(t) = P(Y > t)$. For $i = 1, \ldots, n$:

   B.1. Compute $\hat{S}_{0,b}\left(t|X_i^*\right)$, a nonparametric estimator of the latency $S_0\left(t|X_i^*\right)$, with the original sample. Set $Y_i^* = \infty$ with probability $1 - \hat{p}$, and draw Y_i^* from $\hat{S}_{0,b}(t|X_i^*)$ with probability $\hat{p}$.

   B.2. Generate C_i^* from a nonparametric estimator of $G\left(t|X_i^*\right)$ with the original sample.

   B.3. Compute $T_i^* = \min\left(Y_i^*, C_i^*\right)$ and $\delta_i^* = \mathbf{1}\left(Y_i^* \leq C_i^*\right)$.

C. With the bootstrap resample $\left\{\left(X_i^*, T_i^*, \delta_i^*\right) : i = 1, \ldots, n\right\}$ compute $\hat{\eta}_i^*$ for $i = 1, \ldots, n$.

D. With $\left\{\left(\hat{\eta}_i^*, X_i^*\right) : i = 1, \ldots, n\right\}$, compute the bootstrap versions of U_n in (10) and the corresponding CM and KS statistics, CM_n^* and KS_n^*.

E. Repeat Steps A-D above B times in order to generate B values of the CM and KS statistics, $\left\{CM_n^{*1}, \ldots, CM_n^{*B}\right\}$ and $\left\{KS_n^{*1}, \ldots, KS_n^{*B}\right\}$.

F. The p-value of the CM (respectively, KS) test is approximated as the proportion of values $\left\{CM_n^{*1}, \ldots, CM_n^{*B}\right\}$ larger than CM_n (respectively, $\left\{KS_n^{*1}, \ldots, KS_n^{*B}\right\}$ larger than KS_n).

Note that nonparametric estimators of the conditional functions $S_0(t|x)$ and $G(t|x)$ are required in Step B. Following López-Cheda et al. (2020), if X is continuous, then $S_0(t|x)$ and $G(t|x)$ are estimated with the nonparametric estimator in (3) and the generalized PL estimator in Beran (1981), respectively, and with the corresponding stratified unconditional estimators otherwise.

4 The npcure package: structure and functionality

The **npcure** package provides several functions to model nonparametrically survival data with a possibility of cure. Table 1 contains a compact summary of the available functions. The estimators of the cure rate and latency functions, discussed in the section "Nonparametric estimation in mixture cure models", are implemented by `probcure()` and `latency()`, respectively. The functions `probcurehboot()` and `latencyhboot()` compute bootstrap bandwidths for these estimators. Another function deserving mention in this context is `beran()`, which computes the generalized PL estimator of the conditional survival function $S(t|x)$. A CV bandwidth for use with `beran()` is returned by `berancv()`. Given the computational burden of the procedures implemented by the aforementioned functions, all of them make extensive use of compiled C code. The significance test introduced in the previous section is carried out by `testcov()`, and `testmz()` performs the nonparametric test of Maller and Zhou (1992). Next, a detailed account of the usage of all these functions is provided.

The estimation functions in **npcure** are restricted to one-dimensional continuous covariates. The Epanechnikov kernel is used in the smoothing procedures. Nonparametric estimation with discrete or

Function	Description
`beran`	Computes Beran's estimator of the conditional survival function.
`berancv`	Computes the CV bandwidth for Beran's estimator of the conditional survival function.
`controlpars`	Sets the control parameters of the `latencyhboot()` and `probcurehboot()` functions.
`hpilot`	Computes pilot bandwidths for the nonparametric estimators of the cure rate and the latency.
`latency`	Computes the nonparametric estimator of the latency.
`latencyhboot`	Computes the bootstrap bandwidth for the nonparametric estimator of the latency.
`print.npcure`	Method of the generic function `print` for 'npcure' objects.
`probcure`	Computes the nonparametric estimator of the cure rate.
`probcurehboot`	Computes the bootstrap bandwidth for the nonparametric estimator of the cure rate.
`summary.npcure`	Method of the generic function `summary` for 'npcure' objects.
`testcov`	Performs covariate significance tests for the cure rate.
`testmz`	Performs the nonparametric test of Maller and Zhou (1992).

Table 1: Summary of the functions in the **npcure** package.

categorical variables could be dealt with as in other kernel smoothing procedures. A simple approach is to split the sample into a number of subsets according to the covariate values. When the size of the subsamples is not too small, valid unconditional estimates of the cure probability and latency can be computed. Another alternative is the use of special kernels that can handle any covariate types (see Racine and Li, 2004).

Several features are shared by the functions in the package. All functions return an object of S3 class 'npcure', formally a list of components. Among these components are the primary outputs of the functions, like the computed estimates for `probcure()` and `latency()`, the selected bandwidths for `probcurehboot()` and `latencyhboot()`, or the p-values of the tests for `testcov()` and `testmz()`. The covariate values, observed times, and uncensoring indicators are passed to the functions via the `x`, `t`, and `d` arguments, respectively. Typically, a set of names is passed, which are interpreted as column names of a data frame specified by the `dataset` argument. However, `dataset` may also be left as `NULL`, the default, in which case the objects named in `x`, `t`, and `d` must live in the working directory. More details on these and other arguments are given in the following.

Estimation of the cure rate

The estimation of the cure rate using the nonparametric estimator in (1) is implemented in the `probcure()` function:

```
probcure(x, t, d, dataset = NULL, x0, h, local = TRUE, conflevel = 0L,
   bootpars = if (conflevel == 0 && !missing(h)) NULL else controlpars())
```

The `x0` argument specifies the covariate values where conditional estimates of the cure rate are to be computed. The bandwidths required by the estimator are passed to the `h` argument. The `local` argument is a logical value determining whether the bandwidths are interpreted as local (`local = TRUE`) or global (`local = FALSE`) bandwidths. Notice that if `local = TRUE`, then `h` and `x0` must have the same length. Actually, the `h` argument may be missing, in which case the local bootstrap bandwidth computed by the `probcurehboot()` function is used. This last function implements the procedure for selecting the bandwidth h_x^* described in the section "Bandwidth selection", and its usage is:

```
probcurehboot(x, t, d, dataset, x0, bootpars = controlpars())
```

The `bootpars` argument controls the details of the computation of the bootstrap bandwidth (see section "Bandwidth selection"). In typical use, it is intended to receive the list returned by the `controlpars()` function. The components of this list are described in Table 2.

The function `probcure()` also allows constructing point confidence intervals (CI) for the cure rate. These CIs exploit the asymptotic normality of the estimator (Xu and Peng, 2014), using the bootstrap to obtain an estimate of the standard error of the estimated cure rate. The bootstrap resamples are generated by the same procedure described in the section "Bandwidth selection". Denoting by $z_{1-\alpha/2}$

Argument	Description
B	Number of bootstrap resamples (by default, 999).
hbound	A vector giving the minimum and maximum, respectively, of the initial grid of bandwidths as multiples of the standardized interquartile range (IQR) of the covariate values (by default, c(0.1,3)).
hl	Length of the initial grid of bandwidths (by default, 100).
hsave	A logical specifying if the grid of bandwidths is saved (by default FALSE).
nnfrac	Fraction of the sample size determining the order k of the nearest neighbor used when computing the pilot bandwidth g_x in (7) (by default, 0.25).
fpilot	Either NULL, the default, or a function name. If NULL, the pilot bandwith is computed by the package function hpilot(). If not NULL, it is the name of an alternative, user-defined function for computing the pilot.
qt	In bandwidth selection with latencyhboot(), order of the quantile of the observed times specifying the upper bound of the integral in the computation of the MISE* in (6) (by default, 0.75).
hsmooth	Order of a moving average computed to optionally smooth the selected bandwidths. By default is 1, meaning that no smoothing is done.

Table 2: Summary of the arguments of the controlpars() function.

the $1-\alpha/2$ quantile of a standard normal and by $\widehat{se}_B\left(1-\hat{p}_h(x)\right)$ the estimate of the standard error of $1-\hat{p}_h(x)$ with B bootstrap resamples, a $(1-\alpha)\,100\%$ CI for $1-p(x)$ is computed as:

$$1-\hat{p}_h(x) \mp z_{1-\frac{\alpha}{2}}\widehat{se}_B\left(1-\hat{p}_h(x)\right). \tag{12}$$

The confidence level of the CI is specified through the conflevel argument as a number between 0 and 1. With the special value 0, the default, no CI is computed. Other parameters related to the bootstrap CIs can be passed to the bootpars argument, typically via the output of the controlpars() function. These parameters relate to the number of bootstrap resamples and the computation of the pilot bandwidth, and are specified, respectively, by the B and nnfrac arguments described in Table 2.

The usage of these functions is illustrated with a simulated dataset generated from a model where the cure probability is a logistic function of the covariate:

```
library("npcure")
n <- 50
x <- runif(n, -2, 2)
y <- rweibull(n, shape = 0.5 * (x + 4), scale = 1)
c <- rexp(n, rate = 1)
p <- exp(2 * x)/(1 + exp(2 * x))
u <- runif(n)
t <- ifelse(u < p, pmin(y, c), c)
d <- ifelse(u < p, ifelse(y < c, 1, 0), 0)
data <- data.frame(x = x, t = t, d = d)
```

In the next code example, point and 95% CI estimates of the cure probability are obtained with probcure() at a grid of covariate values ranging from -1.5 to 1.5. For the estimation, the local bootstrap bandwidths previously computed by probcurehboot() are passed to the h argument. The bandwidths, which have been further smoothed with a moving average of 15 bandwidths, are contained in the hsmooth component of the output of probcurehboot(). For the bootstrap, 2000 resamples are generated.

```
x0 <- seq(-1.5, 1.5, by = 0.1)
hb <- probcurehboot(x, t, d, data, x0 = x0,
   bootpars = controlpars(B = 2000, hsmooth = 15))
q1 <- probcure(x, t, d, data, x0 = x0, h = hb$hsmooth, conflevel = 0.95,
   bootpars = controlpars(B = 2000))
q1

#> Bandwidth type: local
#>
#> Conditional cure estimate:
#>         h   x0        cure lower 95% CI upper 95% CI
```

```
#> 0.6212329 -1.5 1.000000000   0.98450759   1.00000000
#> 0.6523881 -1.4 1.000000000   0.87087244   1.00000000
#> 0.6533320 -1.3 1.000000000   0.86080078   1.00000000
#> 0.6606362 -1.2 1.000000000   0.83135572   1.00000000
#> 0.6710717 -1.1 1.000000000   0.82267310   1.00000000
#> 0.6912311 -1.0 0.972213147   0.78259082   1.00000000
#> ...
```

More compactly, the same bootstrap bandwidths would be selected and the same estimates obtained if `h` were left unset when calling `probcure()`:

```
q2 <- probcure(x, t, d, data, x0 = x0, conflevel = 0.95,
   bootpars = controlpars(B = 2000, hsmooth = 15))
```

Figure 1 shows a plot of the true cure rate function and its point and 95% CI estimates at the covariate values saved in `x0`. The plot can be reproduced by executing the next code. The components of the `q1` object accessed by the code are `x0`, keeping the vector of covariate values, `q`, containing the point estimates of the cure rate, and `conf`, a list with the lower (component `lower`) and upper (component `upper`) limits of the CIs for the cure rate.

```
plot(q1$x0, q1$q, type = "l", ylim = c(0, 1), xlab = "Covariate X",
   ylab = "Cure probability")
lines(q1$x0, q1$conf$lower, lty = 2)
lines(q1$x0, q1$conf$upper, lty = 2)
lines(q1$x0, 1 - exp(2 * q1$x0)/(1 + exp(2 * q1$x0)), col = 2)
legend("topright", c("Estimate", "95% CI limits", "True"),
   lty = c(1, 2, 1), col = c(1, 1, 2))
```

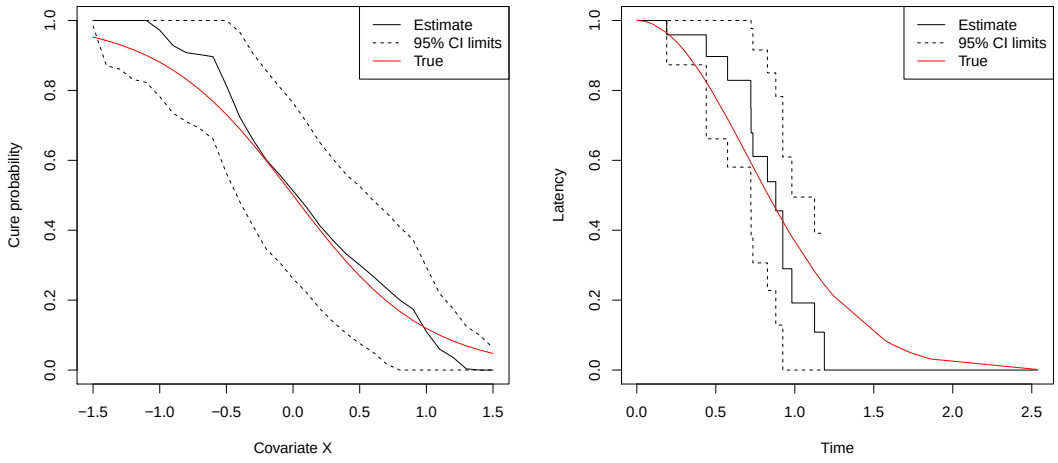

Figure 1: Left panel: estimation of the cure rate. Right panel: estimation of the latency for $x = 0$.

Estimation of the latency function

The latency estimator in (3) is implemented in the `latency()` function:

```
latency(x, t, d, dataset = NULL, x0, h, local = TRUE, testimate = NULL,
  conflevel = 0L, bootpars = if (conflevel == 0) NULL else controlpars(),
  save = TRUE)
```

The function's interface is similar to that of `probcure()`, with all the arguments, except for `testimate`, having exactly the same interpretation. The `testimate` argument determines the times t at which the function $S_0(t|x)$ is estimated. It defaults to `NULL`, which results in the latency being estimated at times given by the `t` argument.

Also, as was the case for `probcure()`, `latency()` allows getting bootstrap CIs for the latency function by specifying their level with the `conflevel` argument. These CIs also rely on the asymptotic

normality of the latency estimator $\hat{S}_{0,b}(t|x)$ in (3) (López-Cheda et al., 2017b). A $(1-\alpha)$ 100% CI for $S_{0,b}(t|x)$ is computed as:

$$\hat{S}_{0,b}(t|x) \mp z_{1-\frac{\alpha}{2}} \widehat{se}_B\left(\hat{S}_{0,b}(t|x)\right), \tag{13}$$

where $\widehat{se}_B\left(\hat{S}_{0,b}(t|x)\right)$ is a bootstrap estimate of the standard error of $\hat{S}_{0,b}(t|x)$, the bootstrap resamples being generated as described in the section "Bandwidth selection".

Also, as with `probcure()`, the user can specify a local or global bandwidth with the combined use of the `h` and `local` arguments. When `h` is left unspecified, a local bootstrap bandwidth is indirectly computed by the `latencyhboot()` function:

```
latencyhboot(x, t, d, dataset = NULL, x0, bootpars = controlpars())
```

This function provides an implementation of the bandwidth selector b_x^* introduced in the section "Bandwidth selection". It is homologous to `probcurehboot()`, with which it shares a common interface. The only noticeable difference is that now the `qt` argument of `controlpars()` (see Table 2) can be used to set u, the upper bound of the integral that must be calculated when computing the bootstrap MISE in (6).

Using the same simulated data as before, the next code illustrates the computation of point and 95% CI estimates (based on 500 bootstrap resamples) of the latency for covariate values 0 and 0.5, and with local bandwidths equal to 0.8 and 0.5, respectively. Notice that, since the `testim` argument is unset, the estimates are computed at the times `t`:

```
S0 <- latency(x, t, d, data, x0 = c(0, 0.5), h = c(0.8, 0.5),
   conflevel = 0.95, bootpars = controlpars(B = 500))
```

To estimate the latency using the bootstrap bandwidth selector, `latencyhbooot()` can be called before calling `latency()`. In the following code, the component `h` of the output of `latencyhbooot()`, where the selected local bandwidths are contained, is passed to the `h` argument of `latency()`:

```
b <- latencyhboot(x, t, d, data, x0 = c(0, 0.5))
S0 <- latency(x, t, d, data, x0 = c(0, 0.5), h = b$h, conflevel = 0.95)
S0
```

```
#> Bandwidth type: local
#>
#> Covariate (x0): 0.0 0.5
#> Bandwidth (h):  4.531978 2.527206
#>
#> Conditional latency estimate:
#>
#> x0 = 0
#>        time   latency lower 95% CI upper 95% CI
#> 0.004599127 1.0000000   1.00000000    1.0000000
#> 0.042088293 1.0000000   1.00000000    1.0000000
#> 0.042271452 1.0000000   1.00000000    1.0000000
#> 0.059671372 1.0000000   1.00000000    1.0000000
#> 0.067375891 1.0000000   1.00000000    1.0000000
#> 0.098569312 1.0000000   1.00000000    1.0000000
#> ...
#>
#> x0 = 0.5
#>        time   latency lower 95% CI upper 95% CI
#> 0.004599127 1.0000000   1.00000000    1.0000000
#> 0.042088293 1.0000000   1.00000000    1.0000000
#> 0.042271452 1.0000000   1.00000000    1.0000000
#> 0.059671372 1.0000000   1.00000000    1.0000000
#> 0.067375891 1.0000000   1.00000000    1.0000000
#> 0.098569312 1.0000000   1.00000000    1.0000000
#> ...
```

An alternative, more succinct way to proceed is to leave `h` unset, since in that case, `latencyhboot()` is indirectly called:

```
S0 <- latency(x, t, d, data, x0 = c(0, 0.5), conflevel = 0.95)
```

Figure 1 shows the estimated and true latencies for covariate value $x = 0$. Next, the code to obtain the plot is reproduced, and it is helpful in illustrating the structure of the output list returned

by `latency()`. The `testim` component has the times at which the estimates are computed. The `S` component is a list having a named item for each covariate value. Each element contains the latency estimates for a covariate value, and the name is constructed from the covariate value by prefixing it with an `x`. The `conf` component is also a named list, the names being constructed as those of the `S` component. Each one of these items contains, structured as a list, the lower (`lower` component) and upper (`upper` component) limits of the CIs. Finally, `x0` keeps the covariate values as a separate element.

```
plot(S0$testim, S0$S$x0, type = "s", xlab = "Time", ylab = "Latency",
   ylim = c(0, 1))
lines(S0$testim, S0$conf$x0$lower, type = "s", lty = 2)
lines(S0$testim, S0$conf$x0$upper, type = "s", lty = 2)
lines(S0$testim, pweibull(S0$testim, shape = 0.5 * (S0$x0[1] + 4),
   scale = 1, lower.tail = FALSE), col = 2)
legend("topright", c("Estimate", "95% CI limits", "True"),
   lty = c(1, 2, 1), col = c(1, 1, 2))
```

Significance test for the cure rate

The **npcure** package also provides an implementation of the nonparametric covariate significance tests for the cure rate discussed in the section "Covariate significance tests":

```
testcov(x, t, d, dataset = NULL, bootpars = controlpars(), save = FALSE)
```

The `x` argument is the covariate whose effect on the cure rate is to be tested. The function's output is a list whose main components are `CM` and `KS`. Each of them, in turn, is a list containing the test statistic (`stat`) and p-value (`pvalue`) of the CM and KS tests, respectively.

The result of the test carried out with our simulated data and 2500 bootstrap resamples is:

```
testcov(x, t, d, data, bootpars = controlpars(B = 2500))
```

```
#> Covariate test
#>
#> Covariate:  x
#>               test statistic p.value
#>   Cramer-von Mises 0.4537077  0.0592
#> Kolmogorov-Smirnov 1.2456568  0.0708
```

Non-numeric covariates can also be tested. For example, for `z`, a nominal covariate added to the simulated data, the result is:

```
data$z <- rep(factor(letters[1:5]), each = 10)
testcov(z, t, d, data, bootpars = controlpars(B = 2500))
```

```
#> Covariate test
#>
#> Covariate:  z
#>               test statistic p.value
#>   Cramer-von Mises 0.2513218  0.6356
#> Kolmogorov-Smirnov 0.7626470  0.5340
```

Estimation of the conditional survival function

The **npcure** package also includes the `beran()` function, which computes the generalized PL estimator of the conditional survival function, $S(t|x)$, by Beran (1981). The `beran()` function in our package may be used together with the `berancv()` function:

```
berancv(x, t, d, dataset, x0, cvpars = controlpars())
```

This function computes the local CV bandwidth selector of Geerdens et al. (2017). It can be directly called by the user, but in practical work should be more usual an indirect call from the `beran()` function, which, as said before, computes the generalized PL estimator of $S(t|x)$:

```
beran(x, t, d, dataset, x0, h, local = TRUE, testimate = NULL, conflevel = 0L,
   cvbootpars = if (conflevel == 0 && !missing(h)) NULL else controlpars())
```

The arguments of these two functions have the same meaning as their homonyms in the `latency()` and `latencyhboot()` functions, `cvpars` and `cvbootpars` playing the role of `bootpars` in these last functions. As in `latency()`, if no bandwidth is provided by the user via `h`, then the local CV bandwidth in Geerdens et al. (2017) is computed by `berancv()`.

For example, the code below computes the Beran estimator for the covariate values 0 and 0.5 using local CV bandwidths. The default behavior of `berancv()` is modified by the auxiliary function `controlpars()`. In detail, the local CV bandwidth search is performed in a grid of bandwidths, which is saved (`hsave = TRUE`) and consists of 200 bandwidths (`hl = 200`) ranging from 0.2 to 2 times the standardized IQR of the covariate (`hbound = c(0.2,2)`). Point and 95% CI estimates of the conditional survival function $S(t|x)$ are computed by `beran()` with the selected bandwidths:

```
x0  <- c(0, 0.5)
hcv <- berancv(x, t, d, data, x0 = x0,
   cvpars = controlpars(hbound = c(0.2, 2), hl = 200, hsave = TRUE))
S <- beran(x, t, d, data, x0 = x0,  h = hcv$h, conflevel = 0.95)
S
```

```
#> Bandwidth type: local
#>
#> Covariate (x0): 0.0 0.5
#> Bandwidth (h):  1.598875 1.104106
#>
#> Beran's conditional survival estimate:
#>
#> x0 = 0
#>        time  survival lower 95% CI upper 95% CI
#> 0.004599127 1.0000000    1.0000000    1.0000000
#> 0.042088293 1.0000000    1.0000000    1.0000000
#> 0.042271452 1.0000000    1.0000000    1.0000000
#> 0.059671372 1.0000000    1.0000000    1.0000000
#> 0.067375891 1.0000000    1.0000000    1.0000000
#> 0.098569312 1.0000000    1.0000000    1.0000000
#> ...
#>
#> x0 = 0.5
#>        time  survival lower 95% CI upper 95% CI
#> 0.004599127 1.0000000    1.0000000    1.0000000
#> 0.042088293 1.0000000    1.0000000    1.0000000
#> 0.042271452 1.0000000    1.0000000    1.0000000
#> 0.059671372 1.0000000    1.0000000    1.0000000
#> 0.067375891 1.0000000    1.0000000    1.0000000
#> 0.098569312 1.0000000    1.0000000    1.0000000
#> ...
```

The next code shows an equivalent way of obtaining the same estimates:

```
S <- beran(x, t, d, data, x0 = x0, conflevel = 0.95,
   cvbootpars = controlpars(hbound = c(0.2, 2), hl = 200, hsave = TRUE))
```

Figure 2 displays point and 95% CI estimates of the survival curve for covariate value 0.5. It has been obtained by executing:

```
plot(S$testim, S$S$x0.5, type = "s", xlab = "Time", ylab = "Survival",
   ylim = c(0, 1))
lines(S$testim, S$conf$x0.5$lower, type = "s", lty = 2)
lines(S$testim, S$conf$x0.5$upper, type = "s", lty = 2)
p0 <- exp(2 * x0[2])/(1 + exp(2 * x0[2]))
lines(S$testim, 1 - p0 + p0 * pweibull(S$testim,
   shape = 0.5 * (x0[2] + 4), scale = 1, lower.tail = FALSE), col = 2)
legend("topright", c("Estimate", "95% CI limits", "True"),
   lty = c(1, 2, 1), col = c(1, 1, 2))
```

Test for enough follow-up

The nonparametric estimators of the cure rate and latency functions given in (1) and (3), respectively, require assumption (4) for their consistency. In other words, the follow-up must be long enough for

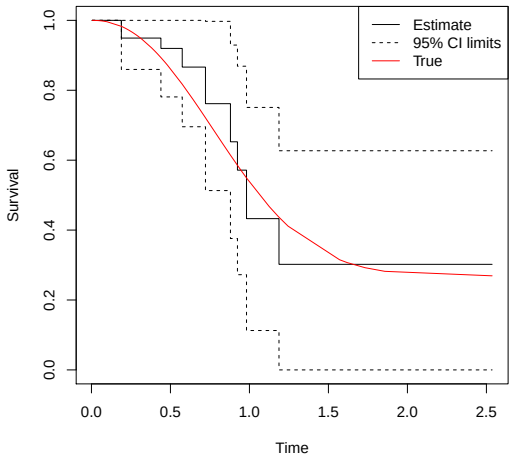

Figure 2: Beran's estimate of the conditional survival function for $x = 0.5$.

cures to happen so that the censored times after the largest uncensored observation can be assumed to correspond to cured subjects.

The procedure to test the hypothesis (4) proposed by Maller and Zhou (1992) is performed by the `testmz()` function:

```
testmz(t, d, dataset)
```

The function returns a list (with class attribute 'npcure') whose main component, containing the p-value of the test, is `pvalue`. The further component `aux` is, in turn, a list of components `statistic`, which contains the test statistic, `n`, the sample size, `delta`, giving the difference between the largest observed time $T_{(n)}$ and the largest uncensored time $T^1_{\max}$, and `interval`, which has the range between $\max(0, T^1_{\max} -$ `delta`$)$ and $T^1_{\max}$.

With our simulated data, the result of the test is:

```
testmz(t, d, data)
```

```
#> Maller-Zhou test
#>
#> statistic  n      p.value
#>        43 50 2.024892e-43
```

5 Example

To illustrate the nonparametric modeling of the mixture cure model with the **npcure** package, we consider the bone marrow transplantation data in Klein and Moeschberger (2005), available as the `bmt` dataset of the R package **KMsurv** (Klein et al., 2012). The data comes from a multi-center study carried out between 1984 and 1989, involving 137 patients with acute myelocytic leukemia (AML) or acute lymphoblastic leukemia (ALL), aged from 7 to 52. Bone marrow transplant (BMT) is the standard treatment for acute leukemia. Transplantation can be considered a failure when leukemia recurs or the patient dies. Consequently, the failure time is defined as the time (days) to relapse or death. The variables collecting this information are:

- `t2` Disease-free survival time in days (time to relapse, death, or end of study)
- `d3` Disease-free survival indicator (`1`: Dead or relapsed, `0`: Alive and disease-free)

The probability of cure after BMT is high, especially if BMT is performed while the patient remains in the chronic phase (Devergie et al., 1987). Recovery after BMT is a complex process depending on a large set of risk factors, whose status is coded by the following variables:

`ta`	Time to acute graft-versus-host disease (GVHD).
`tc`	Time to chronic GVHD.
`tp`	Time to return of platelets to normal levels.
`z1`	Patient age (years).
`z2`	Donor age (years).
`z7`	Waiting time to transplant (days).
`group`	Disease group (`1`: ALL, `2`: AML low risk, `3`: AML high risk).
`da`	Acute GVHD indicator (`1`: Developed, `0`: Never developed).
`dc`	Chronic GVHD indicator (`1`: Developed, `0`: Never developed).
`dp`	Platelet recovery indicator (`1`: Returned to normal, `0`: Never returned to normal).
`z3`	Patient gender (`1`: Male, `0`: Female).
`z4`	Donor gender (`1`: Male, `0`: Female).
`z5`	Patient cytomegalovirus (CMV) status (`1`: Positive, `0`: Negative).
`z6`	Donor CMV status (`1`: Positive, `0`: Negative).
`z8`	FAB (`1`: FAB grade 4 or 5 and AML, `0`: Otherwise).
`z9`	Hospital (`1`: Ohio State University, `2`: Alferd, `3`: St. Vincent, `4`: Hahnemann).
`z10`	Methotrexate (MTX) used for prophylaxis of GVHD (`1`: Yes, `0`: No).

Before applying the estimation methods of the **npcure** package, it should be checked whether the follow-up time was long enough to make it sure that condition (4) holds. This can be subjectively assessed by visualizing a plot of the KM estimate of the unconditional survival function, $S(t)$. The estimated survival curve in Figure 3 suggests the existence of a non-zero asymptote at the right tail. The test of Maller and Zhou (1992) confirms that the follow-up period is adequate to ensure the validity of the nonparametric estimation procedures available in the package:

```
data("bmt", package = "KMsurv")
testmz(t2, d3, bmt)

#> Maller-Zhou test
#>
#> statistic   n      p.value
#>        11 137 1.047242e-05
```

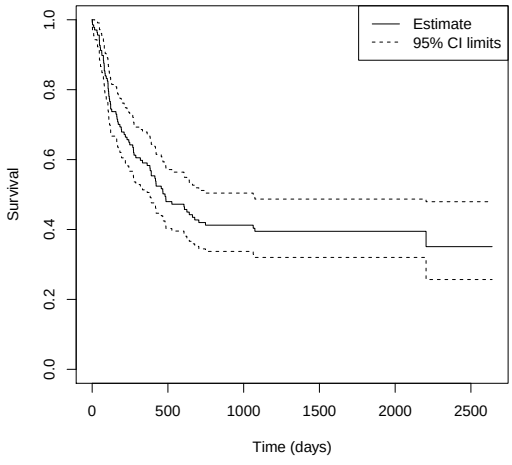

Figure 3: Estimated disease-free survival.

Estimation of the probability of cure

We start by estimating the cure probability as a function of age (`z1`) and waiting time to transplant (`z7`), respectively. Cure probabilities are estimated at a grid of 100 values between the 5th and 95th quantiles of the values of `z1` and `z7`. The code for `z1` is (for `z7`, it is similar):

```
x0 <- seq(quantile(bmt$z1, 0.05), quantile(bmt$z1, 0.95), length.out = 100)
q.age <- probcure(z1, t2, d3, bmt, x0 = x0, conflevel = 0.95,
   bootpars = controlpars(hsmooth = 10))
```

Both estimated cure rates are displayed in Figure 4, where a kernel estimate of the covariate density has been added for reference:

```
par(mar = c(5, 4, 4, 5) + 0.1)
plot(q.age$x0, q.age$q, type = "l", ylim = c(0, 1),
   xlab = "Patient age (years)", ylab = "Cure probability")
lines(q.age$x0, q.age$conf$lower, lty = 2)
lines(q.age$x0, q.age$conf$upper, lty = 2)
par(new = TRUE)
d.age <- density(bmt$z1)
plot(d.age, xaxt = "n", yaxt = "n", xlab = "", ylab = "", col = 2,
   main = "", zero.line = FALSE)
mtext("Density", side = 4, col = 2, line = 3)
axis(4, ylim = c(0, max(d.age$y)), col = 2, col.axis = 2)
legend("topright", c("Estimate", "95% CI limits", "Covariate density"),
   lty = c(1, 2, 1), col = c(1, 1, 2), cex = 0.8)
```

The cure probability seems to be nearly constant or, at most, to decrease slightly with patient age and as the waiting time to transplant increases.

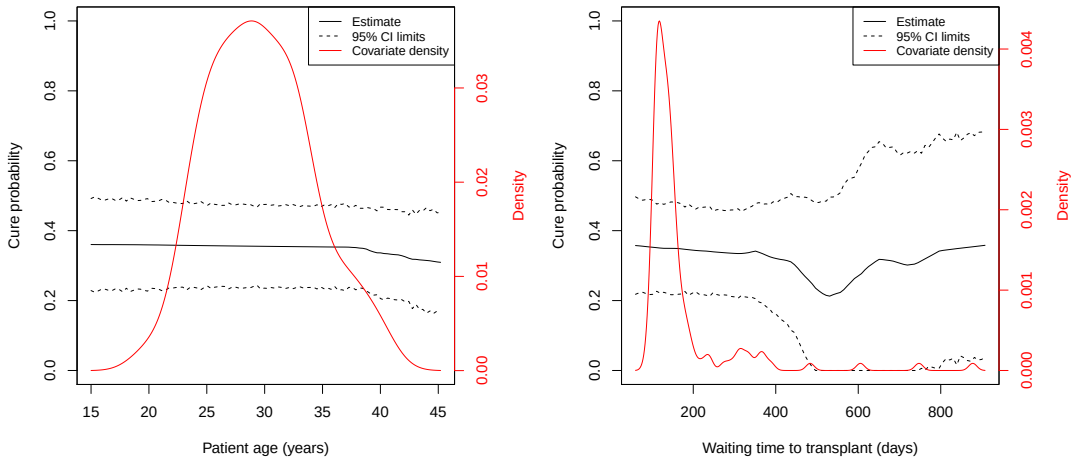

Figure 4: Estimation of the cure probability conditional on age (left panel) and waiting time to transplant (right panel). Nonparametric estimates of the covariate density are plotted for reference.

Testing the effect of one covariate on the probability of cure

The significance of the effect of patient age (`z1`) and waiting time to transplant (`z7`) on the probability of cure can be tested with the `testcov()` function:

```
testcov(z1, t2, d3, bmt,  bootpars = controlpars(B = 2500))
```

```
#> Covariate test
#>
#> Covariate:  z1
#>                test statistic p.value
#>   Cramer-von Mises 0.1103200  0.8204
#> Kolmogorov-Smirnov 0.7308477  0.7900
```

```
testcov(z7, t2, d3, bmt,  bootpars = controlpars(B = 2500))
```

```
#> Covariate test
#>
#> Covariate:  z7
#>                test statistic p.value
#>   Cramer-von Mises 0.7921912  0.0968
#> Kolmogorov-Smirnov 1.6116129  0.1008
```

The effect of age on the cure probability is not statistically significant with neither the CM nor the KS tests ($p_{CM} = 0.820$ and $p_{KS} = 0.790$, where the subscripts identify the p-value in an obvious way). As for the effect of waiting time to transplant, it reaches a borderline significance ($p_{CM} = 0.097$ and $p_{KS} = 0.101$).

Cure probability can also be compared between groups defined by a categorical covariate. We illustrate this case by considering gender (`z3`) and the use of MTX for prophylaxis of GVHD (`z10`). For improving readability, we first label the groups:

```
bmt$z3 <- factor(bmt$z3, labels = c("Male", "Female"))
bmt$z10 <- factor(bmt$z10, labels = c("MTX", "No MTX"))
summary(bmt[, c("z3", "z10")])
```

```
#>      z3         z10
#> Male  :57   MTX   :97
#> Female:80   No MTX:40
```

The estimated survival functions are displayed in Figure 5. The code for gender (`z3`) is:

```
library("survival")
Sgender <- survfit(Surv(t2, d3) ~ z3, data = bmt)
Sgender
```

```
#> Call: survfit(formula = Surv(t2, d3) ~ z3, data = bmt)
#>
#>           n events median 0.95LCL 0.95UCL
#> z3=Male   57     36    318     172      NA
#> z3=Female 80     47    606     418      NA
```

```
plot(Sgender, col = 1:2, mark.time = FALSE, xlab = "Time (days)",
   ylab = "Disease-free survival")
legend("topright", legend = c("Male", "Female"), title = "Gender",
   lty = 1, col = 1:2)
```

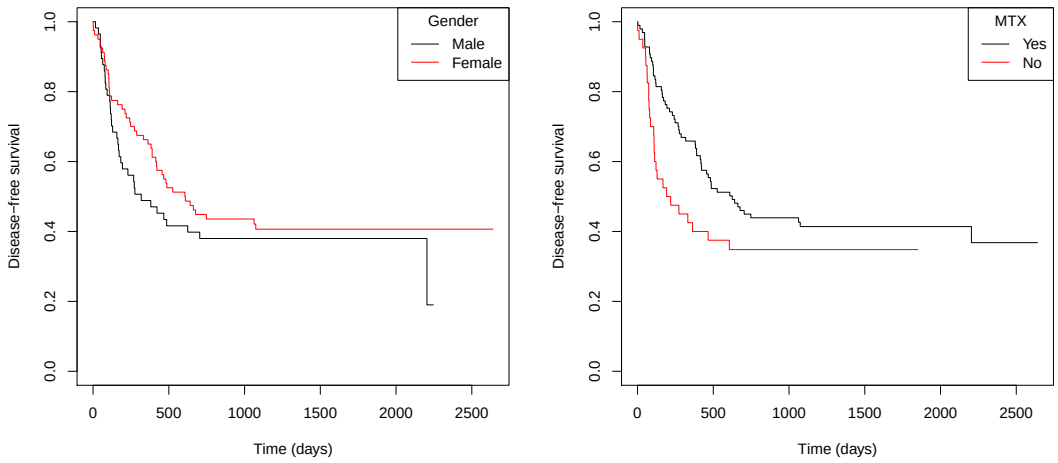

Figure 5: Survival curves of patients conditional on gender (left panel) and use of MTX for prophylaxis of GVHD (right panel).

The estimated probability of cure for each group defined by gender (`z3`) is obtained by computing for each stratum the unconditional cure rate estimator of Laska and Meisner (1992). This estimator of the probability of cure is the value of the KM curve at $T^1_{\max}$ (i.e., it is the minimum of the KM estimate):

```
qgender <- c(min(Sgender[1]$surv), min(Sgender[2]$surv))
qgender
```

```
#> [1] 0.1899671 0.4065833
```

The estimated probability of cure is 19.0% for males and 40.7% for females. The cure probabilities according to the use or not of MTX as GVHD prophylactic (`z10`) are:

```
Smtx <- survfit(Surv(t2, d3) ~ z10, data = bmt)
qmtx <- c(min(Smtx[1]$surv), min(Smtx[2]$surv))
qmtx

#> [1] 0.3679977 0.3482143
```

The cure rate of patients treated with MTX is estimated to be 36.8%, slightly higher than 34.8%, the estimate for patients not treated with MTX.

The effect of these two binary variables on the cure probability is tested with the `testcov()` function similarly as it was done with continuous covariates:

```
testcov(z3, t2, d3, bmt, bootpars = controlpars(B = 2500))

#> Covariate test
#>
#> Covariate:  z3
#>                test statistic p.value
#>   Cramer-von Mises 0.5947305  0.0900
#> Kolmogorov-Smirnov 1.1955919  0.0892

testcov(z10, t2, d3, bmt, bootpars = controlpars(B = 2500))

#> Covariate test
#>
#> Covariate:  z10
#>               test statistic p.value
#>   Cramer-von Mises  1.018441  0.0692
#> Kolmogorov-Smirnov  1.199340  0.0668
```

The differences in the probability of cure between males and females, and between patients with and without MTX treatment are not statistically significant, although a borderline effect is evidenced ($p_{CM} = 0.090$ and $p_{KS} = 0.089$ for gender, $p_{CM} = 0.069$ and $p_{KS} = 0.067$ for MTX).

Estimation of the latency function

The survival of the uncured patients (latency) is estimated for patient age (`z1`) 25 and 40 years as follows:

```
S0 <- latency(z1, t2, d3, bmt, x0 = c(25, 40), conflevel = 0.95,
   bootpars = controlpars(B = 500))
```

Figure 6 displays the survival functions for the two ages, obtained by executing:

```
plot(S0$testim, S0$S$x25, type = "s", ylim = c(0, 1),
   xlab = "Time (days)", ylab = "Latency")
lines(S0$testim, S0$conf$x25$lower, type = "s", lty = 2)
lines(S0$testim, S0$conf$x25$upper, type = "s", lty = 2)
lines(S0$testim, S0$S$x40, type = "s", col = 2)
lines(S0$testim, S0$conf$x40$lower, type = "s", lty = 2, col = 2)
lines(S0$testim, S0$conf$x40$upper, type = "s", lty = 2, col = 2)
legend("topright", c("Age 25: Estimate", "Age 25: 95% CI limits",
   "Age 40: Estimate", "Age 40: 95% CI limits"), lty = 1:2,
   col = c(1, 1, 2, 2))
```

An increased survival of younger patients can be observed, but the survival advantage vanishes after approximately 6 years.

6 Summary

This paper introduces the **npcure** package. It provides an R implementation of a completely nonparametric approach for estimation in mixture cure models, along with a nonparametric covariate significance test for the cure probability. Moreover, the generalized PL estimator of the conditional survival function with a CV bandwidth selection function is included. Furthermore, the theory underlying the implemented methods, presented in Xu and Peng (2014), López-Cheda et al. (2017a), and López-Cheda et al. (2017b), has been compiled.

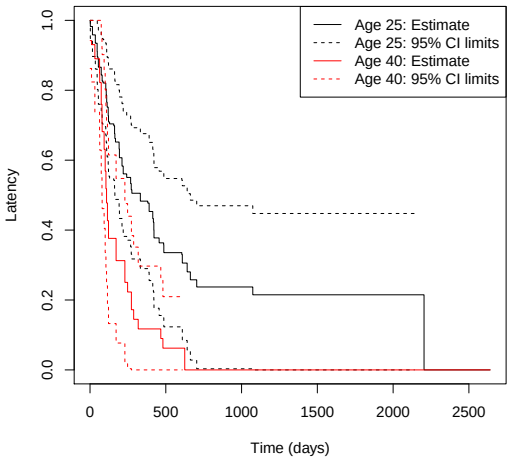

Figure 6: Latency curves of uncured patients 25 and 40 years old.

The **npcure** package has some limitations. Firstly, it only handles right-censored survival times. Left-censored data, truncation, or interval-censored data have not been considered in this approach, and it remains an open problem to be dealt with in the future. Secondly, a conditional estimation can be performed when only one covariate is involved. The same restriction applies to the implemented covariate significance test for the cure rate. An important extension would be the development of estimation and test procedures for the cure rate and latency functions when they depend on a set of covariates. A major challenge is the way the covariates are handled. In that case, the analysis of a large number of covariates would suffer from the curse of dimensionality. Dimension reduction techniques would be required, which leads to a demanding approach that has not been addressed yet, and we leave for further research.

There is an interesting issue that remains an open problem to be dealt with in future versions of the package. Traditional cure rate models implicitly assume that there is no additional information on the cure status of the patients. So, the cure indicator is modeled as a latent variable. However, examples contradicting this assumption can be found. For instance, in some clinical settings, subjects who are followed up beyond a threshold period without experiencing the event can be considered as cured. In other cases, complementary diagnostic tests providing further information about a patient's cure status may be available. We aim to develop improved non-parametric methods of estimation and hypothesis testing that take into account this additional information.

7 Acknowledgments

The first author's research was sponsored by the Beatriz Galindo Junior Spanish Grant from Ministerio de Ciencia, Innovación y Universidades (MICINN) with reference BGP18/00154. All the authors acknowledge partial support by the MICINN Grant MTM2017-82724-R (EU ERDF support included), and by Xunta de Galicia (Centro Singular de Investigación de Galicia accreditation ED431G/01 2016-2019 and Grupos de Referencia Competitiva CN2012/130 and ED431C2016-015) and the European Union (European Regional Development Fund - ERDF).

Bibliography

J. Amdahl. *flexsurvcure: Flexible Parametric Cure Models*, 2017. URL https://CRAN.R-project.org/package=flexsurvcure. R package version 0.0.2. [p21]

M. Amico and I. Van Keilegom. Cure models in survival analysis. *Annual Review of Statistics and Its Application*, 5:311–342, 2018. URL https://doi.org/10.1146/annurev-statistics-031017-100101. [p21]

R. Beran. *Nonparametric Regression with Randomly Censored Survival Data*. University of California, Berkeley, 1981. [p22, 23, 25, 26, 31]

A. Bertrand, C. Legrand, and I. Van Keilegom. *miCoPTCM: Promotion Time Cure Model with Mis-Measured Covariates*, 2020. URL https://CRAN.R-project.org/package=miCoPTCM. R package version 1.1. [p22]

J. W. Boag. Maximum likelihood estimates of the proportion of patients cured by cancer therapy. *Journal of the Royal Statistical Society B*, 11:15–53, 1949. URL https://doi.org/10.1111/j.2517-6161.1949.tb00020.x. [p21]

J. Brettas. *intercure: Cure Rate Estimators for Interval Censored Data*, 2016. URL https://CRAN.R-project.org/package=intercure. R package version 0.1.0. [p22]

M. Brockmann, T. Gasser, and E. Herrmann. Locally adaptive bandwidth choice for kernel regression estimators. *Journal of the American Statistical Association*, 88:1302–1309, 1993. URL https://doi.org/10.2307/2291270. [p25]

C. Cai, Y. Zou, Y. Peng, and J. Zhang. *smcure: Fit Semiparametric Mixture Cure Models*, 2012. URL https://CRAN.R-project.org/package=smcure. R package version 2.0. [p21]

C. Cai, S. Wang, W. Lu, and J. Zhang. *NPHMC: Sample Size Calculation for the Proportional Hazards Mixture Cure Model*, 2013. URL https://CRAN.R-project.org/package=NPHMC. R package version 2.2. [p21]

R. Cao and W. González-Manteiga. Bootstrap methods in regression smoothing. *Journal of Nonparametric Statistics*, 2:379–388, 1993. URL https://doi.org/10.1080/10485259308832566. [p25]

D. Dabrowska. Variable bandwidth conditional Kaplan-Meier estimate. *Scandinavian Journal of Statistics*, 19:351–361, 1992. [p22]

M. A. Delgado and W. González-Manteiga. Significance testing in nonparametric regression based on the bootstrap. *Annals of Statistics*, 29:1469–1507, 2001. URL https://doi.org/10.1214/aos/1013203462. [p25, 26]

J. W. Denham, E. Denham, K. B. Dear, and G. V. Hudson. The follicular non-Hodgkin's lymphomas - I. the possibility of cure. *European Journal of Cancer*, 32:470–479, 1996. URL https://doi.org/10.1016/0959-8049(95)00607-9. [p21]

A. Devergie, E. Gluckman, F. Varrin, J. L. Huret, J. Meletis, H. D. Castro, D. Bombail, E. Vilmer, R. Traineau, and M. Boiron. La greffe de moelle osseuse allogénique dans la leucémie myéloïde chronique. *Nouvelle Revue Française d'Hématologie*, 29:69–72, 1987. [p33]

V. T. Farewell. The use of mixture models for the analysis of survival data with long-term survivors. *Biometrics*, 38:1041–1046, 1982. URL https://doi.org/10.2307/2529885. [p21]

V. T. Farewell. Mixture models in survival analysis: Are they worth the risk? *Canadian Journal of Statistics*, 14:257–262, 1986. URL https://doi.org/10.2307/3314804. [p21]

A. Gannoun, J. Saracco, and K. Yu. Comparison of kernel estimators of conditional distribution function and quantile regression under censoring. *Statistical Modelling*, 7:329–344, 2007. URL https://doi.org/10.1177/1471082X0700700404. [p22]

C. Geerdens, E. F. Acar, and P. Janssen. Conditional copula models for right-censored clustered event time data. *Biostatistics*, 19:247–262, 2017. URL https://doi.org/10.1093/biostatistics/kxx034. [p22, 25, 31, 32]

T. A. Gerds. *prodlim: Product-Limit Estimation for Censored Event History Analysis*, 2018. URL https://CRAN.R-project.org/package=prodlim. R package version 2018.04.18. [p22]

X. Han, Y. Zhang, and Y. Shao. *rcure: Robust Cure Models for Survival Analysis*, 2017. URL https://CRAN.R-project.org/package=rcure. R package version 0.1.0. [p22]

M. C. Iglesias-Pérez. Comparación de dos selectores de la ventana en la estimación de la distribución condicional con censura. In SGAPEIO, editor, *Proceedings of the IX Congreso Galego de Estatística e Investigación de Operacións*. Sociedade Galega para a Promoción da Estatística e Investigación de Operacións, 2009. URL http://sidor.uvigo.es/ixsgapeio/resumenes/81_29_paper.pdf. [p22]

J. D. Kalbfleisch and R. L. Prentice. *The Statistical Analysis of Failure Time Data*. Wiley, New York, 2nd edition, 2002. ISBN 978-0-471-36357-6. [p21]

E. L. Kaplan and P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53:458–481, 1958. URL https://doi.org/10.2307/2281868. [p22]

J. P. Klein and M. L. Moeschberger. *Survival Analysis: Techniques for Censored and Truncated Data*. Springer-Verlag, New York, 2nd edition, 2005. ISBN 978-0-387-21645-4. [p33]

J. P. Klein, M. L. Moeschberger, and J. Yan. *KMsurv: Data sets from Klein and Moeschberger (1997), Survival Analysis*, 2012. URL https://CRAN.R-project.org/package=KMsurv. R package version 0.1-5. [p33]

A. Y. C. Kuk and C. H. Chen. A mixture model combining logistic regression with proportional hazards regression. *Biometrika*, 79:531–541, 1992. URL https://doi.org/10.1093/biomet/79.3.531. [p21]

E. M. Laska and M. J. G. Meisner. Nonparametric estimation and testing in a cure model. *Biometrics*, 48:1223–1234, 1992. URL https://doi.org/10.2307/2532714. [p22, 23, 36]

C. Li and J. M. G. Taylor. A semi-parametric accelerated failure time cure model. *Statistics in Medicine*, 21:3235–3247, 2002. URL https://doi.org/10.1002/sim.1260. [p21]

G. Li and S. Datta. A bootstrap approach to nonparametric regression for right censored data. *Annals of the Institute of Statistical Mathematics*, 53:708–729, 2001. URL https://doi.org/10.1023/A:1014644700806. [p24, 25]

A. López-Cheda, R. Cao, M. A. Jácome, and I. Van Keilegom. Nonparametric incidence estimation and bootstrap bandwidth selection in mixture cure models. *Computational Statistics & Data Analysis*, 105:144–165, 2017a. URL https://doi.org/10.1016/j.csda.2016.08.002. [p22, 23, 24, 25, 37]

A. López-Cheda, M. A. Jácome, and R. Cao. Nonparametric latency estimation for mixture cure models. *TEST*, 26:353–376, 2017b. URL https://doi.org/10.1007/s11749-016-0515-1. [p22, 23, 24, 25, 30, 37]

A. López-Cheda, M. A. Jácome, I. Van Keilegom, and R. Cao. Nonparametric covariate hypothesis tests for the cure rate in mixture cure models. *Statistics in Medicine*, 39:2291–2307, 2020. URL https://doi.org/10.1002/sim.8530. [p25, 26]

I. López-de-Ullibarri, A. López-Cheda, and M. A. Jácome. *npcure: Nonparametric Estimation in Mixture Cure Models*, 2020. URL https://CRAN.R-project.org/package=npcure. R package version 0.1-5. [p22]

R. A. Maller and S. Zhou. Estimating the proportion of immunes in a censored sample. *Biometrika*, 79:731–739, 1992. URL https://doi.org/10.1093/biomet/79.4.731. [p23, 24, 26, 27, 33, 34]

R. A. Maller and S. Zhou. *Survival Analysis with Long-Term Survivors*. Wiley, Chichester, U. K., 1996. URL https://doi.org/10.1002/cbm.318. [p21]

L. Meira-Machado and M. Sestelo. condsurv: An R package for the estimation of the conditional survival function for ordered multivariate failure time data. *The R Journal*, 8(2):460–473, 2016. URL https://doi.org/10.32614/RJ-2016-059. [p22]

L. Meira-Machado, M. Sestelo, and G. Soutinho. *survidm: Inference and Prediction in an Illness-Death Model*, 2019. URL https://CRAN.R-project.org/package=survidm. R package version 1.2.0. [p22]

U. U. Müller and I. Van Keilegom. Goodness-of-fit tests for the cure rate in a mixture cure model. *Biometrika*, 106:211–227, 2019. URL https://doi.org/10.1093/biomet/asy058. [p21, 25]

Y. Niu and Y. Peng. *geecure: Marginal Proportional Hazards Mixture Cure Models with Generalized Estimating Equations*, 2018. URL https://CRAN.R-project.org/package=geecure. R package version 1.0-6. [p22]

Y. Peng. *mixcure: Mixture Cure Models*, 2020. URL https://CRAN.R-project.org/package=mixcure. R package version 2.0. [p22]

Y. Peng and K. B. Dear. A nonparametric mixture model for cure rate estimation. *Biometrics*, 56:237–243, 2000. URL https://doi.org/10.1111/j.0006-341X.2000.00237.x. [p21]

Y. Peng, K. B. Dear, and J. W. Denham. A generalized F mixture model for cure rate estimation. *Statistics in Medicine*, 17:813–830, 1998. URL https://doi.org/10.1002/(SICI)1097-0258(19980430)17:8<813::AID-SIM775>3.0.CO;2-%23. [p21]

J. Racine and Q. Li. Nonparametric estimation of regression functions with both categorical and continuous data. *Journal of Econometrics*, 119:99–130, 2004. URL https://doi.org/10.1016/S0304-4076(03)00157-X. [p27]

J. Xu and Y. Peng. Nonparametric cure rate estimation with covariates. *Canadian Journal of Statistics*, 42:1–17, 2014. URL https://doi.org/10.1002/cjs.11197. [p22, 23, 24, 27, 37]

J. Zhou, J. Zhang, and W. Lu. Computationally efficient estimation for the generalized odds rate mixture cure model with interval-censored data. *Journal of Computational and Graphical Statistics*, 27: 48–58, 2017. URL https://doi.org/10.1080/10618600.2017.1349665. [p22]

Ana López-Cheda
Research Group MODES, CITIC, Departamento de Matemáticas, Facultade de Informática, Universidade da Coruña
CITIC, Campus de Elviña s/n, A Coruña 15071
Spain
(ORCiD: 0000-0002-3618-3246)
ana.lopez.cheda@udc.es

M. Amalia Jácome
Research Group MODES, CITIC, Departamento de Matemáticas, Facultade de Ciencias, Universidade da Coruña
Rúa da Fraga s/n, A Zapateira, A Coruña 15071
Spain
(ORCiD: 0000-0001-7000-9623)
maria.amalia.jacome@udc.es

Ignacio López-de-Ullibarri
Research Group MODES, Departamento de Matemáticas, Escuela Universitaria Politécnica, Universidade da Coruña
15405, Ferrol, A Coruña
Spain
(ORCiD: 0000-0002-3438-6621)
ignacio.lopezdeullibarri@udc.es