1-1-2013

Advanced Methodology Developments in Mixture Cure Models

Chao Cai
University of South Carolina

Advanced Methodology Developments in Mixture Cure Models

by

Chao Cai

Bachelor of Engineering
Nanjing Forestry University 2001
Master of Art
University of Pittsburgh 2003

Submitted in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy in

Biostatistics

The Norman J. Arnold School of Public Health

University of South Carolina

2013

Accepted by:

Jiajia Zhang, Major Professor

Jim Hussey, Committee Member

Bo Cai, Committee Member

Tim Hanson, Committee Member

Lacy Ford, Vice Provost and Dean of Graduate Studies

Acknowledgments

This dissertation would not have been possible without the guidance of my committee members, and support from my family.

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Jiajia Zhang, who has been instrumentally helpful and offered invaluable advice. Her guidance on my dissertation has allowed me to enhance my learning in the area of mixture cure models. Sincere gratitude is also given to my committee members, Dr. Jim Hussey, Dr. Bo Cai, and Dr. Tim Hanson, for their advices and encouragements to make my dream into reality.

I would also like to thank Dr. Yingwei Peng, Dr. Wenbin Lu, Dr. Yubo Zou and Dr. Songfeng Wang for their reviews and comments on my papers. I also want to express my sincere gratitude to all other faculty members and students in the Department of Epidemiology and Biostatistics for sharing their knowledge, enthusiasm and passion in Biostatistics with me over years.

In addition, I would also like to acknowledge National Cancer Institute for the financial support on my dissertation work.

Last, yet importantly, I'm always indebted to my beloved families, my parents, my husband and my two lovely kids for their endless love and understanding. They've always been standing by and supporting me during my doctoral studies.

Abstract

Modern medical treatments have substantially improved cure rates for many chronic diseases and have generated increasing interest in appropriate statistical models to handle survival data with non-negligible cure fractions. The mixture cure models are designed to model such data set, which assume that the studied population is a mixture of being cured and uncured. In this dissertation, I will develop two programs named `smcure` and `NPHMC` in R. The first program aims to facilitate estimating two popular mixture cure models: the proportional hazards (PH) mixture cure model and accelerated failure time (AFT) mixture cure model. The second program focuses on designing the sample size needed in survival trial with or without cure fractions based on the PH mixture cure model and standard PH model. The two programs have been tested by comprehensive simulation settings and real data analysis. Currently, they are available for download from R CRAN. The third project in my dissertation will focus on developing a new estimation method for the PH mixture cure model with allowing patients to die from other causes. The performance of proposed method has been evaluated by extensive simulation studies.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

One of the most important statistical models in handling survival data is the Cox proportional hazards (PH) model. A common unstated assumption behind this model is that all patients will eventually experience the event of interest, given that the follow-up time is long enough. However, with the development of medical studies, more and more fatal diseases are now curable. Therefore, in some clinical studies, a substantial proportion of patients may never experience the event because the treatment has effectively cured the patients. Statistically speaking, an estimated Kaplan-Meier survival curve will tend to level off at a value greater than 0 after a certain time. That is, after sufficient follow-up, the survival curve will reach a plateau.

We refer to these subjects who never experience the event as cured (nonsusceptible) and the remaining subjects as uncured (susceptible). The main interests of such data are to determine the proportion of cured patients, the failure time distribution of uncured patients, and the possible effects of covariates.

In this chapter, we will introduce two examples of data with cure fractions in Section 1.1. The first data is melanoma data from Eastern Cooperative Oncology Group (ECOG) and the second one is leukemia data from Bone Marrow Transplant study. The two basic survival regression models: the PH model and AFT model will be discussed in Section 1.2. In Section 1.3, we will give the outline of this dissertation.

Eastern Cooperative Oncology Group (ECOG) Data

We consider melanoma data from the ECOG phase III clinical trial E1684 [16]. This study has been investigated by many authors [16, 6, 5, 11, 8]. The aim of the E1684 clinical trial was to evaluate the high dose interferon alpha-2b (IFN) regimen against the placebo as the postoperative adjuvant therapy on relapse-free survival (RFS) in patients with American Joint Committee on Cancer (AJCC) stage IIB or III melanoma.

A total of 287 patients with high-risk melanomas were accrued to E1684 between 1984 and 1990. High-risk patients were defined to include those designated as stage IIB or III by the former AJCC staging system (primary tumor >4 mm depth with or without regional lymph node involvement or shallower lesions with pathologically proven lymphatic metastases or regional lymph node recurrence). Patients were treated with wide local excision and complete regional lymph node dis-section and then randomized to adjuvant high dose IFN (20 MU/m IV 5 days per week for 4 weeks, followed by 10 MU/m 3 days per week SC for 48 weeks) or observation group. The results of this trial were first reported in 1996 with a median follow-up time of 6.9 years (range, 0.6 to 9.6 years) [16]. After deleting 2 observations with missing data, analysis of treatment effects versus observation group was based on data from 285 patients randomized to IFN or observation group in trial E1684. The median RFS was 1.721 years in the IFN arm versus 0.982 year in the observation arm with a p-value of 0.0118 by log-rank test. The IFN treatment group is coded as 1 and observation group as 0. The response variable is RFS in years.

The time data is listed in Table 1.1. There were 140 patients in observation group and 145 patients in IFN treatment group. The Kaplan-Meier survival curves for the IFN treatment group and observation group are given in Figure 1.1. The

RFS is significant better for IFN group compared to the observation group (p-value = 0.0118 by log rank test).

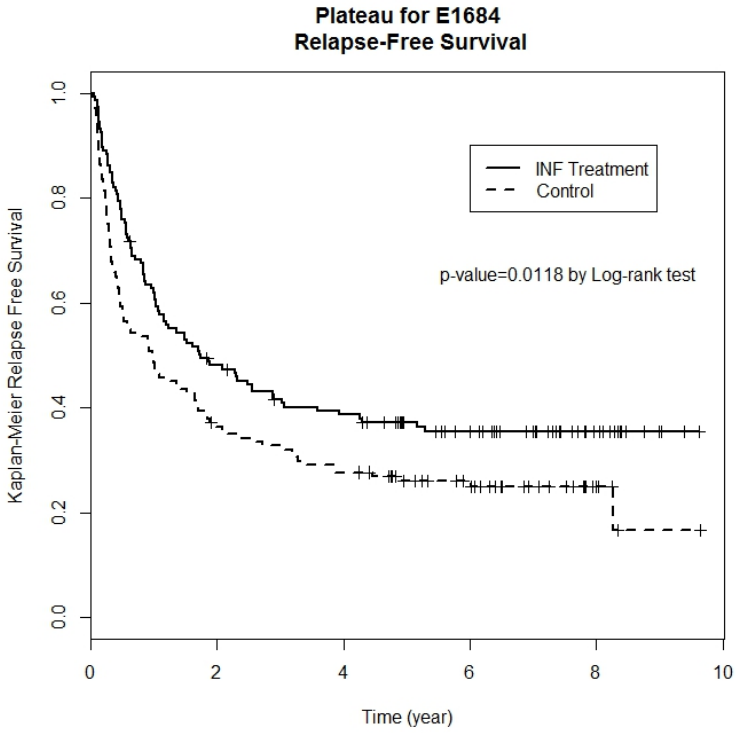

Figure 1.1 Kaplan-Meier relapse-free survival for the IFN treatment group and the control group.

From the Kaplan-Meier survival curves, we can see that the estimated RFS curve from the IFN group is above the observation group, which indicates that the survival probability of patients from the IFN group is higher than that from the observation group. It also shows that both curves level off at a value substantially greater than 0 after about 8-year follow-up, which indicates that some patients will not experience the recurrence after treatments. Therefore, there may exist cured patients in both treatment group and observation group.

Table 1.1 ECOG 1684

High-dose interferon alfa-2b (IFN)	0.04932, 0.06027, 0.09863, 0.10411, 0.11507, 0.12329, 0.13151, 0.13151, 0.14247, 0.14521, 0.15890, 0.17260, 0.17260, 0.18082, 0.18082, 0.19178, 0.24658, 0.26027, 0.26027, 0.26027, 0.30137, 0.30137, 0.33699, 0.34247,0.34795, 0.35616, 0.39178, 0.41370, 0.43836, 0.43836, 0.46027, 0.46027, 0.47945, 0.49041, 0.49041, 0.54795, 0.55068, 0.55068, 0.56164, 0.56986, 0.59178, 0.60822*, 0.62466, 0.63288, 0.64384, 0.64658, 0.70959, 0.78630, 0.82466, 0.82740, 0.82740, 0.85479, 0.85753, 0.86301, 0.95068, 0.99726, 1.00274, 1.00822, 1.02192, 1.03288, 1.07123, 1.09041, 1.15068, 1.16438, 1.19178, 1.23836, 1.35068, 1.47671, 1.48767, 1.52055, 1.61370, 1.70411, 1.72055, 1.73425, 1.83562*, 1.85753, 1.88219, 2.08219, 2.15068*, 2.28219, 2.29589, 2.31507, 2.47671, 2.55068, 2.55616, 2.87397, 2.87671, 2.90411*, 3.02466, 3.05479, 3.58630, 3.93973, 4.26301, 4.29041, 4.29863*, 4.36164*, 4.63836*, 4.81918*, 4.86027*, 4.89315*, 4.90685*, 4.92877*, 4.94795*, 5.16712, 5.30137, 5.45479*, 5.54521*, 5.59178*, 5.75890*, 6.00000*, 6.13699*, 6.20274*, 6.34795*, 6.37808*, 6.41096*, 6.47123*, 6.89315*, 7.00000*, 7.04110*, 7.04384*, 7.23288*, 7.30685*, 7.35616*, 7.41918*, 7.42466*, 7.62192*, 7.70959*, 7.82192*, 7.83562*, 7.96438*, 7.96712*, 8.04110*, 8.09863*, 8.21370*, 8.28767*, 8.33425*, 8.36712*, 8.40000*, 8.45753*, 8.75342*, 8.98630*, 8.99178*, 9.03288*, 9.38356*,9.63014*
Placebo	0.03288, 0.06027, 0.06027, 0.07671, 0.07945, 0.08493, 0.09589, 0.09863, 0.09863, 0.10685, 0.10959, 0.12603, 0.12603, 0.12877, 0.13151, 0.13973, 0.13973, 0.14795, 0.14795, 0.15342, 0.17260, 0.17260, 0.18630, 0.19452, 0.19452, 0.21918, 0.23014, 0.23288, 0.23836, 0.24110, 0.24110, 0.24110, 0.25753, 0.26575, 0.26575, 0.28767, 0.28767, 0.28767, 0.28767, 0.30959, 0.30959, 0.32055, 0.32055, 0.32603, 0.32877, 0.34795, 0.35616, 0.36438, 0.38630, 0.41096, 0.41644, 0.43562, 0.44384, 0.45205, 0.45479, 0.46301, 0.46575, 0.49589, 0.51507, 0.51507, 0.52603, 0.57260, 0.59726, 0.62740, 0.72603, 0.89589, 0.90137, 0.91781, 0.92877, 0.97808, 0.98630, 0.98904, 1.00822, 1.01644, 1.06849, 1.08219, 1.18904, 1.35342, 1.35616, 1.51507, 1.64110, 1.65205, 1.69589, 1.69863, 1.70959, 1.83288, 1.84658, 1.87945, 1.89863*, 1.94795, 2.09041, 2.13151, 2.39726, 2.57808, 2.72329, 3.06849, 3.18630, 3.20548, 3.27671, 3.33973, 3.85205, 3.86575, 4.24384*, 4.40822*, 4.45205, 4.72055*, 4.74795*, 4.76986*, 4.81370*, 4.86849, 4.95616*, 5.12329*, 5.24658*, 5.32329*, 5.77808*, 5.88493*, 6.01096, 6.02192*, 6.06575*, 6.15890*, 6.30959*, 6.40548*, 6.48493*, 6.51507*, 6.85479*, 6.93151*, 7.09589*, 7.25205*, 7.53151*, 7.63836*, 7.79726*, 7.81370*, 7.83288*, 7.94521*, 7.99726*, 8.02740*, 8.24658*, 8.26301, 8.34247*, 9.64384*

Bone Marrow Transplant Study for Leukemia Patients

We consider the bone marrow transplant study for the refractory acute lymphoblastic leukemia patients, which was first analyzed by [15]. This data set is widely used in the AFT mixture cure model because the PH assumption is not appropriate for the latency distribution [31]. Figure 1.2 illustrates the logarithm of the estimated cumulative hazard functions for the uncensored patients for each group based on the Nelson-Aalen estimator. It is easy to see that the two curves cross over which indicate that the PH assumption is not appropriate for this data set.

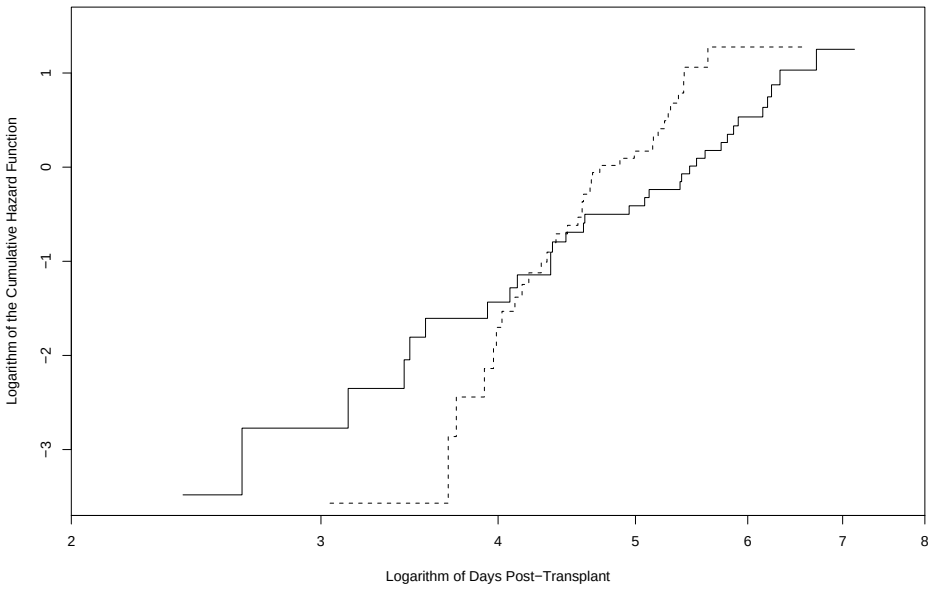

Figure 1.2 Logarithm of the cumulative hazard function curves. Dashed line for the autologous transplant, solid line for the allogeneic transplant.

There were 46 patients in the allogeneic treatment and 44 patients in the autologous treatment group. The treatment variable is included in both incidence and latency parts (1 for autologous treatment group; 0 for allogeneic treatment group). The data set is listed in Table 1.2, and the Kaplan-Meier survival curves for the two treatment groups are given in Figure 1.3.

Table 1.2 Bone marrow transplant treatment of high risk acute lymphoblastic leukemia (March 1982–December 1985, University of Minnesota).

allogeneic treatment group	11, 14, 23, 31, 32, 35, 51, 59, 62, 78, 78, 79, 87, 99, 100, 141, 160, 166, 216, 219, 235, 250, 270, 313, 332, 352, 368, 468, 491, 511, 557, 628*, 726*, 819, 915*, 966*, 1109*, 1158*, 1256, 1614*, 1619*, 1674*, 1712*, 1745*, 1820*, 1825*
autologous treatment group	21, 40, 42, 50, 53, 54, 56, 61, 64, 67, 73, 76, 79, 81, 88, 95, 98, 98, 99, 104, 105, 106, 112, 131, 147, 171, 172, 179, 189, 195, 199, 213, 223, 224, 277, 724*, 729*, 734, 1053*, 1094*, 1192*, 1475*, 1535*, 1535*, 1845*

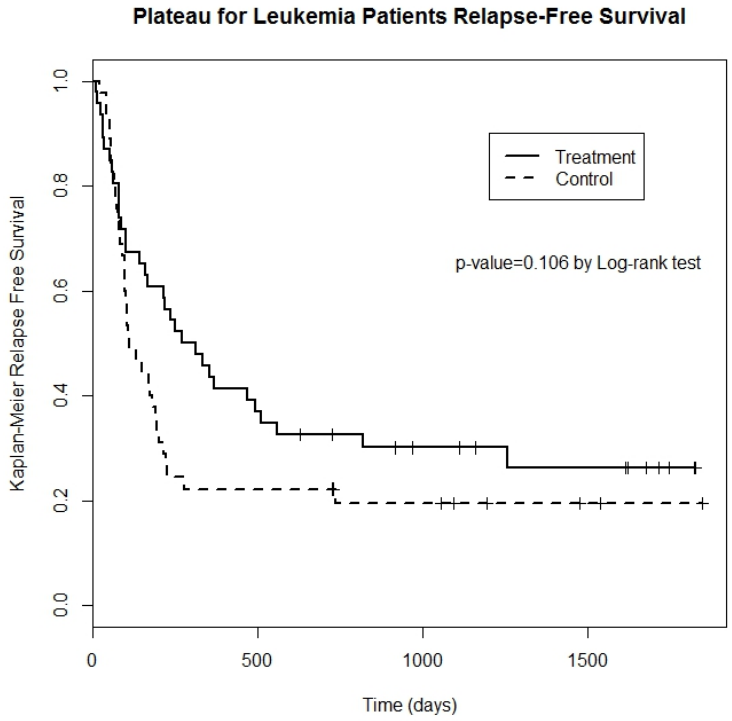

Figure 1.3 Kaplan-Meier survival for Bone Marrow Transplant Study.

From the Kaplan-Meier survival curves, we can see that the estimated survival curve from the allogeneic treatment group is above the one from the autologous treatment group, which indicates that the survival probability of patients from the allogeneic treatment group is higher than that from the autologous treatment group (p-value = 0.106 by log-rank test). It also shows that both curves level off at a value

substantially greater than 0 after one or two years follow-up, which indicates that some patients will not experience the recurrence after the treatments. Therefore, there may exist cured patients in both treatment groups.

Both data display the possible cure fractions, in order to estimate the proportion of cured patients accurately, a cure rate model has to be considered. The most commonly used type of cure rate model is the mixture cure model which was first developed by Boag in 1949 [2] and later developed by Berkson and Gage in 1952 [1]. After that, there are many extensions on the mixture cure model, such as the PH mixture cure model and AFT mixture cure model. In this dissertation, we will focus on the software developments and advanced methodology developments in various mixture cure models.

1.2 Basic Survival Regression Models

Standard PH Model

The PH assumption provides a way to introduce covariates into models and to separate the effect of the covariates and the shape of a baseline hazard function. It has been successfully employed in Cox's PH regression model for survival data. The PH model can be expressed as

$$h(t) = h_0(t)\exp(\boldsymbol{\beta}\mathbf{x}), \tag{1.1}$$

where $h(\cdot)$ is the hazard function, $h_0(\cdot)$ is the baseline hazard function and $\boldsymbol{\beta}$ is a vector of unknown coefficients of interest. If $\mathbf{x} = 0$, the hazard function $h(t)$ is equal to the baseline hazard function $h_0(t)$. The model is called the proportional hazards model because if we look at two individuals with covariate values $\mathbf{x}_1$ and $\mathbf{x}_2$, the ratio of their hazards

$$\frac{h(t|\mathbf{x}_1)}{h(t|\mathbf{x}_2)} = \exp(\boldsymbol{\beta}(\mathbf{x}_1 - \mathbf{x}_2)),$$

is a constant. Therefore, the hazards are proportional. The quantity $\exp(\boldsymbol{\beta}(\mathbf{x}_1 - \mathbf{x}_2))$ is called as the relative risk (hazard ratio) of an individual with risk factor $\mathbf{x}_1$ having the event as compared to an individual with risk factor $\mathbf{x}_2$. In particular, if $\mathbf{x}_1$ indicates the treatment effect ($\mathbf{x}_1 = 1$ if treatment and $\mathbf{x}_1 = 0$ if placebo) and all other covariates have the same value, $\exp(\boldsymbol{\beta})$ is the relative risk between patients in the treatment group and control group with other risks factors fixed at the same value.

The cumulative hazards function is defined as the integral of the baseline hazard function. If $H(t)$ is the cumulative hazards function corresponding to $h(t)$, $H(t) = \int_0^\infty h(t)dt$. Let $H(t)$ and $H_0(t)$ be the cumulative hazards functions corresponding to $h(t)$ and $h_0(t)$. The logarithm of the cumulative hazard function satisfies the following equation

$$\log(H(t)) = \boldsymbol{\beta}\mathbf{x} + \log(H_0(t)).$$

Therefore, the curves of the logarithm of the cumulative hazard function for various levels of $\mathbf{x}$ should be parallel, which is referred to as the PH assumption. Usually, this assumption should be verified before using the PH model.

The main innovation of the PH model is that $\boldsymbol{\beta}$ can be estimated without specifying, or even estimating, a baseline hazard function. This is accomplished by developing the concept of the partial likelihood, a likelihood function which only depends on $\boldsymbol{\beta}$. In the rest of this section, we will introduce the partial likelihood estimation method of the PH model. Let $\mathbf{O} = (t_i, \delta_i, \mathbf{x_i})$ denote the observed data for the ith individual $i = 1, \cdots, n$, where t_i is the observed survival time, δ_i is the censoring indicator with $\delta_i = 1$ for the uncensored time and 0 for the censored time, and $\mathbf{x_i}$ is the value of covariate. We assume that censoring is noninformative in that, given $\mathbf{x}_i$, the event and censoring time for the ith patient are independent.

Suppose that there are no ties among event times. Let $t_1 < t_2 < \cdots < t_D$ denote the ordered event times and $\mathbf{x}_{(i)k}$ be the kth covariate associated with the individual

whose failure time is t_i. Define the risk set at time t_i, $R(t_i)$, as the set of all individuals who are still under study at a time just prior to t_i. The partial likelihood for the PH model is expressed by

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{D} \frac{\exp[\sum_{k=1}^{p} \beta_k \mathbf{x}_{(i)k}]}{\sum_{j \in R(t_i)} \exp[\sum_{k=1}^{p} \beta_k \mathbf{x}_{jk}]} \,.$$

This is treated as a usual likelihood, and inference is carried out by usual means. It is of interest to note that the numerator of likelihood depends only on information from the individual who experiences the event, whereas the denominator utilizes information about all individuals who have not yet experienced the event (including some individuals who will be censored later).

Let $LL(\boldsymbol{\beta}) = \ln[L(\boldsymbol{\beta})]$, we can write $LL(\boldsymbol{\beta})$ as

$$LL(\boldsymbol{\beta}) = \sum_{i=1}^{D} \sum_{k=1}^{p} \beta_k \mathbf{x}_{(i)k} - \sum_{i=1}^{D} \ln \left[\sum_{j \in R(t_i)} \exp \left(\sum_{k=1}^{p} \beta_k \mathbf{x}_{jk} \right) \right].$$

The efficient score equation is found by taking partial derivatives with respect to $\boldsymbol{\beta}$ as follows. Let $U_b(\boldsymbol{\beta}) = \partial LL(\boldsymbol{\beta}) / \partial \beta_b$, $b = 1, \ldots, p$. Then,

$$U_b(\boldsymbol{\beta}) = \sum_{i=1}^{D} \mathbf{x}_{(i)b} - \sum_{i=1}^{D} \frac{\sum_{j \in R(t_i)} \mathbf{x}_{jb} \exp[\sum_{k=1}^{p} \beta_k \mathbf{x}_{jk}]}{\sum_{j \in R(t_i)} \exp[\sum_{k=1}^{p} \beta_k \mathbf{x}_{jk}]}$$

The partial maximum likelihood estimates are found by maximizing $LL(\boldsymbol{\beta})$. This maximization procedure can be done in most statistical software packages, such as `coxph` in R and `PROC PHREG` in SAS.

Standard AFT Model

The AFT model regresses the logarithm of survival time over covariates, which is a useful alternative to the PH model, when the PH assumption does not satisfy. We can specify the AFT model by

$$\log(T_i^*) = \boldsymbol{\beta} \mathbf{x}_i + \varepsilon_i \,, \tag{1.2}$$

where ε_i's are independent random errors and $E(\varepsilon_i)$ may not be zero. One advantage of the AFT model over the PH model is that the covariate effects on the failure time are modeled directly rather than indirectly, as in the PH model. Thus, the interpretation of covariate effects in the AFT model is much simpler than in the PH model.

There are many discussions on parametric estimation methods [18, 13] and semiparametric estimation methods [28, 23] for the AFT model. Our main interest focuses on the semiparametric estimation method. Tsiatis [28] proposed the rank estimation method and Ritov [23] considered the general linear square estimation method. However, Ritov[23] proved the equivalency between the rank estimation method and the general linear square estimation method.

In the rest of this section we will give a brief description of the rank estimation method in the semiparametric AFT model. The rank estimation method can be derived from the partial likelihood principle of the PH model. Consider the usual PH model with the regression coefficient vector γ, say

$$h(\varepsilon_i) = h_0(\varepsilon_i) \exp(\gamma' \mathbf{x}_i),$$

where $\varepsilon_i = \log t_i - \boldsymbol{\beta} \mathbf{x}_i$. The partial log likelihood function is

$$\sum_{i=1}^{n} \delta_i \Big(\gamma' \mathbf{x}_i - \log \sum_{j=1}^{n} e^{\gamma \mathbf{x}_j} I(\varepsilon_j \geq \varepsilon_i) \Big),$$

where $I(\cdot)$ is the indicator function. The derivative of the logarithm of the partial likelihood function with respect to γ is simply:

$$\Psi(\gamma) = \sum_{i=1}^{n} \delta_i \left(\mathbf{x}_i - \frac{\sum_{j=1}^{n} \mathbf{x}_j e^{\gamma \mathbf{x}_j} I(\varepsilon_j \geq \varepsilon_i)}{\sum_{j=1}^{n} e^{\gamma \mathbf{x}_j} I(\varepsilon_j \geq \varepsilon_i)} \right) .$$

If the parameter γ is 0, $\Psi(0) = 0$ can be used as a linear rank estimating equation for $\boldsymbol{\beta}$ (ε_i is a function of $\boldsymbol{\beta}$). It is important to note that as long as the underlying failure times T_i^* are independent and identically distributed, for large n, $\Psi(0)$ is approximately centered around 0. It can also be extended to include a general (predictable)

weight function. That is, we rewrite $\Psi(0)$ as $\Psi(\boldsymbol{\beta}; k(\cdot))$:

$$\Psi(\boldsymbol{\beta}; k(\cdot)) = \sum_{i=1}^{n} \delta_i k(\varepsilon_i) \left(\mathbf{x}_i - \frac{\sum_{j=1}^{n} \mathbf{x}_j I(\varepsilon_j \geq \varepsilon_i)}{\sum_{j=1}^{n} I(\varepsilon_j \geq \varepsilon_i)} \right) ,$$

where $k(\cdot)$ is a general (predictable) weight function. For example, $k(u) = \sum_{j=1}^{n} I(\varepsilon_j \geq u)/n$ is called as the Gehan weight function. Once the weight function is specified, $\Psi(\boldsymbol{\beta}) = 0$ is the estimating equation for $\boldsymbol{\beta}$.

However, the above semiparametric estimating functions are step functions of the regression parameters with potentially multiple roots, and the corresponding estimators may not be well defined. Jin[12] provided simple and reliable methods for implementing the aforementioned rank estimators. They showed that the rank estimator with the Gehan weight function can be readily obtained by minimizing a convex objective function through a standard linear programming technique. Their procedure yielded a consistent root and can be extended to other choices of weight function. Under the Gehan weight function, Jin[12] showed that $\Psi(\boldsymbol{\beta}; k(\cdot))$ can be simplified as

$$n^{-1} \sum_{i=1}^{n} \sum_{j=1}^{n} \delta_i (\mathbf{x}_i - \mathbf{x}_j) I(\varepsilon_j \geq \varepsilon_i) ,$$

which is the gradient of the convex function

$$n^{-1} \sum_{i} \sum_{j} \delta_i \left(\varepsilon_i - \varepsilon_j \right) I(\varepsilon_i \leq \varepsilon_j) .$$

Therefore it can be easily minimized by the linear programming method.

1.3 Outline of Dissertation

In Chapter 2, we will outline the PH mixture cure model and its computational estimation method. In the mixture cure model, the `smcure` package is developed to estimate the semiparametric PH mixture cure model with covariates where the cure fraction can be modeled by various binary regression models and the survival of uncured individuals can be modelled by the PH survival model. The R function

of the `smcure` package and its usage are also described. The results of simulation study are provided to evaluate the performance of the package. An example is given to illustrate the usage of the package.

In Chapter 3, we will focus on the sample size design of a study with possible long-term survivors based on the PH mixture cure model. An R package `NPHMC` will be introduced. This package is developed to facilitate physicians or clinicians to design a study with or without cure fraction based on the semiparametric PH mixture cure model or standard PH model. The parameters of sample size formula can be specified based on previous literature reviews or estimated based on historical or observed data via `smcure` R package.

In Chapter 4, we will propose a new estimation method for PH mixture cure model allowing other causes of death. with competing risks data. The model and its computational method will be discussed. The results of simulation study and application to real data analysis will also be provided.

In Chapter 5, the cure rate model will be extended to the AFT mixture cure model. When the PH assumption does not satisfy for the uncured patients, an AFT model is an alternative to model the latency party of mixture cure model. The estimation method and its application in R will also be discussed.

Chapter 6 is the summary and conclusions of mixture cure models. Some future work will also be discussed.

Chapter 2

Estimating Semiparametric PH Mixture Cure Model and Software Program Development

2.1 Abstract

The mixture cure model is a special type of survival models and it assumes that the studied population is a mixture of uncure (susceptible) individuals who may experience the event of interest, and cure (non-susceptible) individuals who will never experience the event. For such data, standard survival models are usually not appropriate because they do not account for the possibility of cure. The mixture model has been widely used in medical research. The aim of this chapter is to present the PH mixture cure model and an R package `smcure`, which fits the PH mixture cure model semiparametrically.

2.2 Introduction

The PH model is the most popular model in survival analysis. As stated before, the common unstated assumption behind this model is that all patients will eventually experience the event of interest, given that the follow-up time is long enough. However, in recent years, with the development of medical studies, more and more fetal diseases are now cured. There has been an increasing interest in modelling survival data with long term survivors. Such data often arise from clinical trials. Thus, there is a need to develop statistical models to analyze whether the treatment can cure the disease or slow down the progression of the disease if not cure.

The mixture cure model, firstly introduced by Boag (1949) [2] and Berkson and Gage (1952)[1], is one of the popular models to estimate the cure rate of the treatment and the survival rate of uncure patients at the same time.

Let T denote the failure time of interest, $1-\pi(\mathbf{z})$ be the probability of a patient being cured depending on $\mathbf{z}$, and $S(t|\mathbf{x})$ be the survival probability of the uncured patients depending on $\mathbf{x}$, where $\mathbf{x}$ and $\mathbf{z}$ are observed values of two covariate vectors that may affect the survival function. The mixture cure model can be expressed as

$$S_{pop}(t|\mathbf{x},\mathbf{z}) = \pi(\mathbf{z})S(t|\mathbf{x}) + 1 - \pi(\mathbf{z}) \tag{2.1}$$

Usually, $\pi(\mathbf{z})$ is refer to as "incidence" and $S(t|\mathbf{x})$ is refer to as "latency". If the PH model is used to model the latency part, the cure model is called the PH mixture cure model.

An R package called `smcure` is developed to estimate semiparametric PH mixture cure model and semiparametric AFT mixture cure model. This chapter will only focus on PH mixture cure model and AFT mixture cure model will be discussed in Chapter 5.

In section 2.3, we outline models and the computational methods. The R function and its arguments are described in Section 2.4. Simulation results are displayed in Section 2.5. We use an example to illustrate the `smcure` package in Section 2.6.

2.3 Model and Computational Method

Semiparametric PH Mixture Cure Model

An advantage of the mixture cure model is that the proportion of cured subjects and the survival distribution of uncured subjects are modeled separately and the interpretation of effects of $\mathbf{x}$ and $\mathbf{z}$ is straightforward.

Usually, a logit link function

$$\pi(\mathbf{z}) = \frac{\exp(\mathbf{bz})}{1+\exp(\mathbf{bz})},$$

where $\mathbf{b}$ is a vector of unknown parameters, is used to model the effects of $\mathbf{z}$. Other link functions can also be applied to the incidence part, such as the complementary log-log link

$$\log(-\log(1-\pi(\mathbf{z}))) = \mathbf{bz},$$

and the probit link

$$\Phi^{-1}(\pi(\mathbf{z})) = \mathbf{bz},$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard normal distribution. The logit link is a default option in the `smcure` package.

If the distribution of failure time of uncured patients/latency part can be modeled by a PH model, the mixture cure model is called as the PH mixture cure model.

To model the effect of covariates $\mathbf{x}$ on the failure time distribution of uncured patients in the mixture model, we employ the PH assumption to model the effect of $\mathbf{x}$ on the distribution by $h(t) = h_0(t)\exp(\boldsymbol{\beta}\mathbf{x})$. This assumption implies that $S(t|\mathbf{x}) = S_0(t)^{\exp(\boldsymbol{\beta}\mathbf{x})}$ where $S_0(t)$ is the baseline survival function of uncured subjects when $\mathbf{x} = 0$. Parametric approaches to the mixture cure model were studied by many authors [10, 21, 30]. Since it is usually difficult to verify a parametric assumption, there has been increasing interest in the semiparametric mixture cure models [19, 20, 26, 27, 31]. This chapter will focus on semiparametric estimation method for the PH mixture cure model.

Computational Method

Let $\mathbf{O} = (t_i, \delta_i, \mathbf{z_i}, \mathbf{x_i})$ denote the observed data for the ith individual $i = 1, \cdots, n$, where $\mathbf{z_i}, \mathbf{x_i}$ are the possible covariates in the incidence and latency parts respectively. We assume that the censoring is independent and noninformative. It is worthwhile

pointing out that the same covariates are allowed for the incidence and latency components although we use different covariate notations for these two components.

Let $\mathbf{\Theta} = (\mathbf{b}, \boldsymbol{\beta}, S_0(t))$ denote the unknown parameters. To use the EM algorithm to estimate unknown parameters in this PH mixture cure model, let y_i be an indicator of cure status of the ith patient, namely, $y_i = 1$ if the patient is uncured and 0 otherwise, $i = 1, 2, \cdots, n$.. Obviously, if $\delta_i = 1$, $y_i = 1$; if $\delta_i = 0$, y_i is not observable and it can be one or zero. Note that $\pi(\mathbf{z}) = P(y_i = 1|\mathbf{z})$. Let $\mathbf{y} = (y_1, y_2, \cdots, y_n)$. Therefore, $\mathbf{y}$ is partially missing information which will be employed in the EM algorithm.

Given $\mathbf{y} = (y_1, y_2, \cdots, y_n)$ and $\mathbf{O}$, the complete likelihood function can be expressed as

$$\prod_{i=1}^{n} [1 - \pi(\mathbf{z_i})]^{1-y_i} \pi(\mathbf{z_i})^{y_i} h(t_i|Y = 1, \mathbf{x_i})^{\delta_i y_i} S(t_i|Y = 1, \mathbf{x_i})^{y_i} \tag{2.2}$$

where $h(\cdot)$ is the hazard function corresponding to $S(\cdot)$. The logarithm of the complete likelihood function can be written as $l_c(\mathbf{b}, \boldsymbol{\beta}; \mathbf{O}, \mathbf{y}) = l_{c_1}(\mathbf{b}; \mathbf{O}, \mathbf{y}) + l_{c_2}(\boldsymbol{\beta}; \mathbf{O}, \mathbf{y})$, where

$$l_{c_1}(\mathbf{b}; \mathbf{O}, \mathbf{y}) = \sum_{i=1}^{n} y_i \log[\pi(\mathbf{z_i})] + (1 - y_i) \log[1 - \pi(\mathbf{z_i})], \tag{2.3}$$

$$l_{c_2}(\boldsymbol{\beta}; \mathbf{O}, \mathbf{y}) = \sum_{i=1}^{n} y_i \delta_i \log[h(t_i|Y = 1, \mathbf{x_i})] + y_i \log[S(t_i|Y = 1, \mathbf{x_i})]. \tag{2.4}$$

The E-step in the EM algorithm computes the conditional expectation of the complete log-likelihood with respect to $y_i's$, given the observed data $\mathbf{O}$ and current estimates of parameters $\mathbf{\Theta^{(m)}} = (\mathbf{b}^{(m)}, \boldsymbol{\beta}^{(m)}, S_0^{(m)}(t))$. The conditional expectation of y_i will be enough to complete this step since both (2.3) and (2.4) are linear functions of y_i. The expectation of $E(y_i|\mathbf{O}, \mathbf{\Theta^{(m)}})$ can be written as

$$w_i^{(m)} = E(y_i|\mathbf{O}, \mathbf{\Theta^{(m)}}) = \delta_i + (1 - \delta_i) \frac{\pi(\mathbf{z}_i) S(t_i|Y = 1, \mathbf{x_i})}{1 - \pi(\mathbf{z_i}) + \pi(\mathbf{z}_i) S(t_i|Y = 1, \mathbf{x_i})} \Bigg|_{(\mathbf{O}, \mathbf{\Theta^{(m)}})}. \tag{2.5}$$

It is easy to see that $w_i^{(m)} = 1$ if $\delta_i = 1$ and $w_i^{(m)}$ is the probability of uncured patients if $\delta_i = 0$. Thus, the second part of $E(y_i|\mathbf{O}, \mathbf{\Theta^{(m)}})$ can be interpreted as the

conditional probability of the ith individual remaining uncured. Because $\delta_i \log w_i^{(m)} = 0$ and $\delta_i w_i^{(m)} = \delta_i$, the expectations of (2.3) and (2.4) can be written as

$$E(l_{c_1}) = \sum_{i=1}^{n} w_i^{(m)} \log[\pi(\mathbf{z_i})] + (1 - w_i^{(m)}) \log[1 - \pi(\mathbf{z_i})], \tag{2.6}$$

$$E(l_{c_2}) = \sum_{i=1}^{n} \delta_i \log[w_i^{(m)} h(t_i|Y = 1, \mathbf{x_i})] + w_i^{(m)} \log[S(t_i|Y = 1, \mathbf{x_i})]. \tag{2.7}$$

The M-step in the EM algorithm is to maximize (2.6) and (2.7) with respect to the unknown parameters. The parameters in equation (2.6) can be easily estimated by '`glm`' package in R. Because the expressions of equation (2.7) and $w_i^{(m)}$ depend on the latency assumption, we will first demonstrate the estimation approach under the PH mixture cure model.

Peng and Dear [22] and Sy and Taylor [26] proposed a partial likelihood type method to estimate $\boldsymbol{\beta}$ without specifying the baseline hazard function. The estimating equation (2.7) can be written as

$$\log \prod_{i=1}^{n} [h_0(t_i) \exp(\boldsymbol{\beta}\mathbf{x}_i + \log(w_i^{(m)}))]^{\delta_i} S_0(t_i)^{\exp(\boldsymbol{\beta}\mathbf{x}_i + \log(w_i^{(m)}))}, \tag{2.8}$$

which is similar to the log-likelihood function of the standard PH model with the additional offset variable $\log(w_i^{(m)})$. Therefore, the parameters in equation (2.7) can be estimated by '`coxph`' package in R. A detailed presentation can be found in Peng [20], Peng and Dear [22], and Sy and Taylor [26].

Estimation of the Survival Function in the E-Step

In the E-step, we update w_i by (2.5). This updating involves the survival function $S(t|Y = 1)$, which also involves the baseline survival function $S_0(t|Y = 1)$ for given $\hat{\boldsymbol{\beta}}$. Therefore, the estimation method of baseline survival function $S_0(t|Y = 1)$ based on the current information is needed to complete the E-step.

Let $t_{(1)} < t_{(2)} < \cdots < t_{(k)}$ be the distinct uncensored failure times, $d_{t_{(j)}}$ denote the number of events and $R(t_{(j)})$ denote the risk set at time $t_{(j)}$. The Breslow-type

estimator for $S_0(t|Y=1)$ is given by

$$\hat{S}_0(t|Y=1) = \exp\left(-\sum_{j:t_{(j)}\leq t} \frac{d_{t_{(j)}}}{\sum_{i\in R(t_{(j)})} w_i^{(m)} e^{\beta \mathbf{x_i}}}\right). \tag{2.9}$$

Because the estimator, $\hat{S}_0(t|Y=1)$, may not approach 0 as $t \to \infty$, we set $\hat{S}_0(t|Y=1) = 0$ for $t > t_{(k)}$. Then $\hat{S}(t|Y=1) = \hat{S}_0(t|Y=1)^{\exp(\hat{\boldsymbol{\beta}}\mathbf{x})}$.

Variance Estimation

Because of the complexity of the estimating equation in the EM algorithm, the standard errors of estimated parameters are not directly available. In order to obtain the variance of $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{b}}$, this package uses `sample` function in R to respectively draw random bootstrap samples with replacement from cases and controls. The results of standard errors with different bootstrap sampling numbers for the later two examples are shown in Tables 2.1 and 2.2.

Table 2.1 Eastern Cooperative Oncology Group (ECOG) Data

Cure probability model	SE(nboot=100)	SE(nboot=200)	SE(nboot=500)
Intercept	0.35	0.33	0.29
TRT	0.36	0.33	0.31
SEX	0.34	0.33	0.33
AGE	0.02	0.01	0.01
Failure time distribution model	SE(nboot=100)	SE(nboot=200)	SE(nboot=500)
TRT	0.16	0.17	0.17
SEX	0.18	0.17	0.19
AGE	0.01	0.01	0.01

Table 2.2 Bone Marrow Transplant Study

Cure probability model	SE(nboot=100)	SE(nboot=200)	SE(nboot=500)
Intercept	0.25	0.23	0.26
TRT	0.52	0.48	0.54
Failure time distribution model	SE(nboot=100)	SE(nboot=200)	SE(nboot=500)
Intercept	0.21	0.18	0.18
TRT	0.30	0.27	0.27

2.4 Package Description

The estimation methods discussed above are implemented in the `smcure` package. The `smcure` function in the package can be called with the following syntax:

```
smcure(formula,cureform,offset=NULL,data,na.action=na.omit,
model=c("ph","aft"),link="logit",Var=TRUE,emmax=50,eps=1e-7,nboot=100)
```

The required arguments are:

- `formula`: a formula object, with the response on the left of a '∼' operator, and the variables included in the latency part on the right. The response must be a survival object as returned by the `Surv` function.
- `cureform`: specifies the variables included in the incidence part on the right of a '∼' operator.
- `data`: a data frame containing variables used in `formula` and `cureform`.
- `model`: specifies survival model in the latency component, which can be `"ph"` or `"aft"`.

The optional arguments are:

- `offset`: variable(s) with coefficient 1 in both incidence and latency parts of the semiparametric PH mixture cure model or the semiparametric AFTMC model. By default, `offset = NULL`.
- `na.action`: a missing-data filter function. By default `na.action = na.omit`.
- `link`: specifies the link function in the incidence component. The logit, probit or complementary loglog (cloglog) links are available. By default `link = "logit"`.

- `Var`: if it is `TRUE`, the program returns bootstrap standard errors `Std.Error` for $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{b}}$ by the bootstrap method. If it is set to be `False`, the program only returns coefficient estimates. By default, `Var = TRUE`.

- `emmax`: specifies the maximum iteration number. If the convergence criterion is not met, the EM iteration will be stopped after emmax iterations and the estimates will be based on the last maximum likelihood iteration. The default `emmax = 50`.

- `eps`: sets the convergence criterion. The default is `eps = 1e-7`. The iterations are considered to be converged when the maximum relative change in the parameters and likelihood estimates between iterations is less than the value specified.

- `nboot`: specifies the number of bootstrap samplings. The default `nboot = 100`.

The output is composed of two parts: `Cure probability model` and `Failure time distribution model`. The cure rate can be easily estimated from the output by $1 - \hat{\pi}(\mathbf{z})$. The estimated mixture cure survival function $S_{pop}(\cdot)$ is computed by `predictsmcure` function and plotted by `plotpredictsmcure` function.

Notes of the package:

- The user has to create “dummy variables" outside the package if data has categorical variable with more than two values.

- The “formula" and “cureform" arguments require at least one covariate.

- The default `nboot` = 100 is good number to estimate variance. From the Tables 2.1 and 2.2, we can see that the impact of the choice of 100, 200 and 500 for the `nboot` is trivial.

2.5 Simulation Study

In the simulation study, the probability of cure is generated from a logistic model where $\pi(\mathbf{z}) = \frac{\exp(\mathbf{bz})}{1+\exp(\mathbf{bz})}$. The covariate $\mathbf{z}$ is generated from a binary distribution with a probability of 0.5. Censoring times are generated from a uniform distribution $U(c_1, c_2)$, where constants of c_1 and c_2 are determined to obtain a desired censoring rate. The survival times of uncure patients are generated from either a Weibull distribution where $S(t|Y = 1, \mathbf{z}) = \exp[-(\lambda t)^k \exp(\boldsymbol{\beta}\mathbf{z})]$ with $\lambda = 0.5$ and $k = 1$ or a Lognormal distribution of log N(0,1). The results below are based on $n = 200$ and $n = 500$ respectively with 500 replications.

Table 2.3 Estimates from PHMC model (2,-1,2)

Survival Distribution	Censoring Rate	Parameter	True Values	n = 200 Bias	n = 200 MSE	n = 500 Bias	n = 500 MSE
Weibull	21.9	$\hat{b}_0$	2	0.0644	0.1530	0.0470	0.0505
		$\hat{b}_1$	-1	-0.0590	0.2092	-0.0459	0.0693
		$\hat{\beta}_1$	2	0.0185	0.0514	0.0077	0.0199
	33.4	$\hat{b}_0$	2	-0.4613	0.9757	-0.5052	0.4038
		$\hat{b}_1$	-1	0.4856	1.0264	0.5102	0.4299
		$\hat{\beta}_1$	2	0.0982	0.0328	0.0881	0.0218
Lognormal	21.1	$\hat{b}_0$	2	0.0655	0.1407	0.0318	0.0475
		$\hat{b}_1$	-1	-0.0459	0.1885	-0.0351	0.0675
		$\hat{\beta}_1$	2	0.0238	0.0461	0.0139	0.0190
	29.5	$\hat{b}_0$	2	-0.2903	0.5530	-0.3329	0.2270
		$\hat{b}_1$	-1	0.3022	0.5905	0.3280	0.2429
		$\hat{\beta}_1$	2	-0.0626	0.0584	-0.0695	0.0279

Tables 2.3 and 2.4 present the estimated biases and MSE from the PH mixture cure model of three regression parameters b_0, b_1 and β_1 based on the logistic-Weibull data and logistic-lognormal data.

In Table 2.3, $\boldsymbol{b_0} = 2$ and $\boldsymbol{b_1} = -1$ correspond to $\pi(\mathbf{z} = 0) = 0.88$ and $\pi(\mathbf{z} = 1) = 0.73$ which mean that 12% of the population is cured in the control group and

Table 2.4 Estimates from PHMC model (1.3863,-1,2)

Survival Distribution	Censoring Rate	Parameter	True Values	n = 200 Bias	n = 200 MSE	n = 500 Bias	n = 500 MSE
Weibull	32.1	$\hat{b}_0$	1.3863	0.0638	0.0923	0.0123	0.0291
		$\hat{b}_1$	-1	-0.0566	0.1384	-0.0115	0.0473
		$\hat{\beta}_1$	2	0.0092	0.0615	0.0135	0.0225
	37.5	$\hat{b}_0$	1.3863	0.0714	0.9070	-0.0216	0.0773
		$\hat{b}_1$	-1	-0.0713	0.9791	0.0125	0.0945
		$\hat{\beta}_1$	2	0.0041	0.0667	-0.0024	0.0260
Lognormal	32.0	$\hat{b}_0$	1.3863	0.0265	0.0846	0.0163	0.0325
		$\hat{b}_1$	-1	-0.0209	0.1358	-0.0067	0.0510
		$\hat{\beta}_1$	2	0.0330	0.0594	0.0193	0.0262
	39.4	$\hat{b}_0$	1.3863	-0.1662	0.2517	-0.2246	0.1238
		$\hat{b}_1$	-1	0.1706	0.3061	0.2362	0.1448
		$\hat{\beta}_1$	2	0.0063	0.0807	-0.0613	0.0283

27% in the treatment group. In Table 2.4, $\boldsymbol{b_0} = 1.3863$ and $\boldsymbol{b_1} = -1$ correspond to $\pi(\mathbf{z} = 0) = 0.8$ and $\pi(\mathbf{z} = 1) = 0.6$, which mean that 20% of the population is cured in the control group and 40% in the treatment group.

The simulation results show that estimates of $\boldsymbol{b}$ and $\boldsymbol{\beta}$ do not depend on the assumption of the distribution. The bias of estimates based on the PH mixture cure model are quite small. Even though the bias increase a little bit as the censoring rate increases, the MSE of estimates get smaller as sample size increases from 200 to 500. The same conclusion can be made even we increase cure rates from 12% to 20% in the control group and 27% to 40% in the treatment group as shown in Table 2.4.

Investigate the effect of link functions

As discussed in the section of semiparametric PH mixture cure model, besides 'logit' link, 'probit' link and 'cloglog' can also be used to model the probability of cure.

Suppose we use the same input data that are generated in Table 2.3. We only

consider sample size 500 and Weibull distribution for survival times. We re-estimate the unknown parameters $\boldsymbol{b}$ by 'logit' link, 'probit' link and 'cloglog' link respectively. Then, re-calculate the cure rates for control group and treatment group by different link functions. The estimates of parameters $\boldsymbol{b}$ and cure rates based on 500 replicates are summarized in Table 2.5.

Table 2.5 Estimated cure rates for different link functions (n=500)

PHMC	Survival Distribution	Censoring Distribution	Parameter	True Values	Logit	Probit	Cloglog
	Weibull	U(0.5,9)	$\hat{b}_0$	2	2.0169	1.1463	0.7159
			$\hat{b}_1 = -1$	-1	-1.017	-0.5203	-0.4352
			Cure Rate				
			Control	0.12	0.117	0.206	0.129
			Treatment	0.27	0.269	0.327	0.266

From Table 2.5, we can see that even though the point estimates of unknown parameters $\boldsymbol{b}$ are different, the estimated cure rates by different link functions are quite close.

In order to visually see the effects of different link functions, we plot the estimated uncure rates by 'logit' link, 'probit' link and 'cloglog' link versus a continuous covariate, which takes values ranging from -2 to 2.

From the Figure 2.1 , we can see that the estimated cure rates from 'logit' link, 'probit' link and 'cloglog' are quite close if the input is the same.

2.6 Application

In this section, we use an example to illustrate the use of `smcure` package for the semiparametric PH mixture cure model.

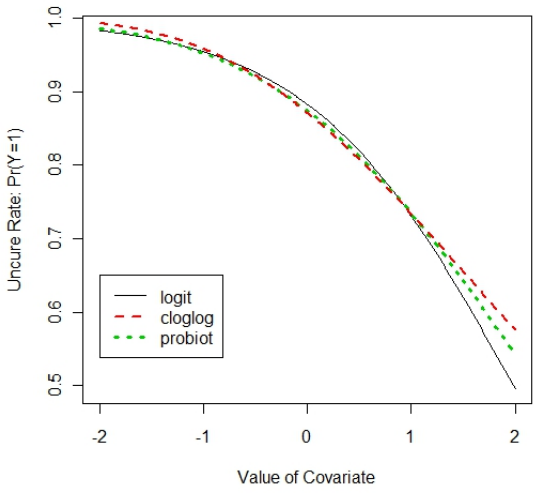

Figure 2.1 Plot of different link functions.

Eastern Cooperative Oncology Group (ECOG) Data

We fit the semiparametric PH mixture cure model to the melanoma data from the ECOG phase III clinical trial E1684 [16], which was also illustrated by `PSPMCM` SAS macro [8]. The aim of the E1684 clinical trial was to evaluate the high dose interferon alpha-2b (IFN) regimen against the placebo as the postoperative adjuvant therapy. After deleting missing data, a total number of 284 observations is used in the analysis. Treatment (0=control,1=treatment), gender (0=male,1=female) and age (continuous variable which is centered to the mean) are used in both the incidence and latency parts. The response variable is relapse free survival in years. The semiparametric PH mixture cure model can be fitted as following:

```
> pd <- smcure(Surv(FAILTIME,FAILCENS)~TRT+SEX+AGE,cureform=~TRT+SEX+AGE,
     data=e1684,model="ph",nboot=500)
```

The output is:

```
> printsmcure(pd)
```

```
Call:
smcure(formula = Surv(FAILTIME, FAILCENS) ~ TRT + SEX + AGE, cureform =
~TRT + SEX + AGE, data = e1684, model = "ph", nboot = 500, Var = TRUE)

Cure probability model:
               Estimate  Std.Error    Z value     Pr(>|Z|)
(Intercept)  1.36493298 0.28769252  4.7444159 2.091088e-06
TRT         -0.58847727 0.30645148 -1.9202951 5.482064e-02
SEX         -0.08696490 0.32905294 -0.2642885 7.915576e-01
AGE          0.02033857 0.01445227  1.4072922 1.593408e-01

Failure time distribution model:
        Estimate   Std.Error    Z value  Pr(>|Z|)
TRT -0.153595097 0.172120117 -0.8923716 0.3721938
SEX  0.099458470 0.190788176  0.5213031 0.6021556
AGE -0.007664013 0.006695195 -1.1447033 0.2523321
```

The standard errors of the estimated parameters are obtained based on 500 bootstrap samples. If considering the male with the median centered age of 0.579, we can draw the fitted survival curves by the treatment group using the following commands:

```
> predm=predictsmcure(pd,newX=cbind(c(1,0),c(0,0),c(0.579,0.579)),
     newZ=cbind(c(1,0),c(0,0),c(0.579,0.579)),model="ph")
> plotpredictsmcure(predm,model="ph")
```

The fitted survival curves for the male with median centered age are shown in Figure 2.2. The upper solid line is the allogeneic treatment group and lower dashed line is the autologous treatment group. The IFN treatment has higher survival probability than the placebo group.

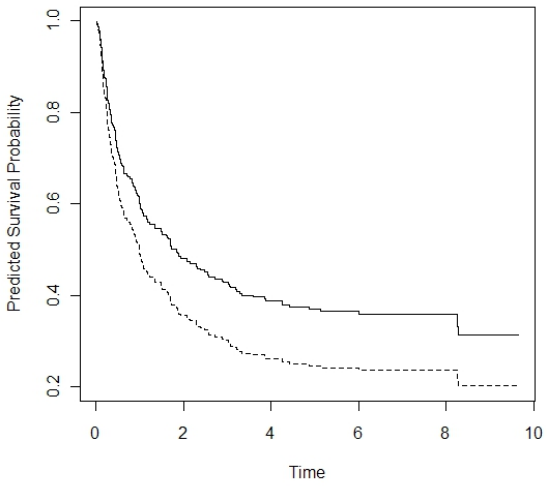

Figure 2.2 Fitted survival curves for the male with median centered age.

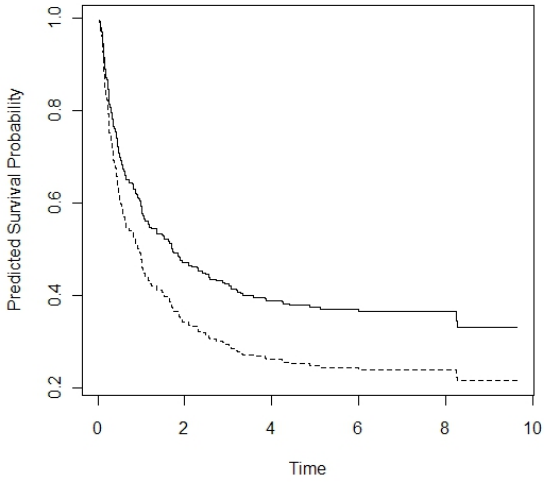

Figure 2.3 Fitted survival curves for the female with median centered age.

```
> predf=predict.smcure(pd,newX=cbind(c(1,0),c(1,1),c(0.579,0.579)),
     newZ=cbind(c(1,0),c(1,1),c(0.579,0.579)),model="ph")
> plotpredictsmcure(predf,model="ph")
```

Similarly, we fitted the survival curves by the treatment group for the female at the same age, which are shown in Figure 2.3. The upper solid line is the IFN treatment and lower dashed line is the control group. The IFN treatment has higher survival probability than the placebo group for female as well.

2.7 Conclusions

We develop an R package to estimate the semiparametric PH mixture cure model. The cure probability part is estimated by the generalized linear model which allows many link functions, such as `logit`, `probit` and `cloglog`. The latency part can follow the PH model. The semiparametric estimation procedures are based on the EM algorithm for both models. The `smcure` package in R is developed for implementing the semiparametric estimation methods to the PH mixture cure model with covariates.

2.8 Availability

The package `smcure` and the relevant documentation can be freely downloaded from CRAN webpage http://cran.r-project.org/package=smcure.

Chapter 3

New Program Development of Sample Size Estimation for PH Mixture Cure Model

3.1 Abstract

Due to the advances in medical research, more and more diseases can be cured nowadays, which largely increases the need for an easy-to-use software in calculating sample size of clinical trials with cure fractions. Current available sample size software, such as `PROC POWER` in SAS, `Survival Analysis` module in PASS, `powerSurvEpi` package in R are all based on the standard proportional hazards (PH) model which is not appropriate to design a clinical trial with cure fractions. Instead of the standard PH model, the PH mixture cure model is an important tool in handling the survival data with possible cure fractions. However, there are no tools available that can help deal with the design of a trial with cure fractions. Therefore, we develop an R package `NPHMC` to determine the sample size needed for such study design.

3.2 Introduction

Sample size calculation is an important component in designing randomized controlled clinical trials with time-to-event endpoints. Assuming constant hazard ratio between the treatment arm and control arm, the following sample size formula based on the standard PH model has been widely used in practice[24, 25]:

$$n = \frac{(Z_{\alpha/2} + Z_\theta)^2}{p(1-p)\beta_0^2 P(\delta = 1)}, \tag{3.1}$$

where α specifies the level of significance of the statistical test and $1-\theta$ specifies the power of the statistical test; $Z_{\alpha/2}$ and Z_θ are the upper $\alpha/2$ and θ percentiles of the standard normal distribution, respectively; p is the proportion of patients being assigned to the treatment arm; β_0 is the log-hazard ratio between treatments; δ is the censoring indicator (1 for failure and 0 for censoring), and $P(\delta = 1)$ is the probability of failure. Assuming $S_C(t) = P(C \geq t)$ is the survival function of the censoring time and $f_0(t)$ is the density function of survival times for uncured patients in the control arm, $P(\delta = 1) = \int_0^\infty S_C(t) f_0(t) dt$. Formula (3.1) has been implemented in most software, and a common assumption is that the baseline density function $f_0(t)$ follows the exponential distribution and the survival function of censoring time $S_C(t)$ is uniform, such as `PROC POWER` in SAS.

One unstated assumption of the standard PH model is that all individuals under study are susceptible to the adverse event of interest, and they would experience the event eventually if there was no censoring. However, more and more patients will be cured nowadays due to the advances in recent medical research, that is, those patients may never experience the event even after a sufficient follow-up period. The mixture cure model [17, 22, 26] is particularly designed to handle the dataset with a cure fraction. Unlike the standard survival model, the mixture cure model has two components in order to model the cure probability and the survival probability of uncured patients.

Assume $S_j^*(\cdot)$ denote the overall survival function, $S_j(\cdot)$ denote the survival function of uncured patients and π_j $(0 \leq \pi_j < 1)$ is the cure rate in arm j, $j = 0$ for control arm and $j = 1$ for treatment arm. The mixture cure model can be written as

$$S_j^*(t) = \pi_j + (1 - \pi_j) S_j(t). \tag{3.2}$$

Specifically, the PH mixture cure model is designed by assuming the PH model for survival probability of uncured patients $S_j(t)$ and logistic regression for the cure probability π_j.

In this chapter, we design an R package `NPHMC` to implement the sample size calculation proposed in [29]. The sample size formula based on the PH mixture cure model (3.2) includes the sample size calculation based on the standard PH model. Thus, the R package `NPHMC` is an extension of the exiting sample size software for designing survival trial. In the next Section, we outline the computational method. The R function and its arguments are described in Section 3.4. A simulation study comparing parametric with nonparametric sample size calculation is discussed in Section 3.5. Two examples are provided to illustrate the usage of the `NPHMC` package in Section 3.6.

3.3 Computational Method

Let T denote the observed times, which is the minimum of the failure time and censoring time. We assume that the censoring is independent. Let $\lambda_j^*(\cdot)$ denote the overall hazard function and $\lambda_j(\cdot)$ denote the hazard function of uncured patients for arm j, $j = 0, 1$ respectively. The PH mixture cure model (3.2) assumes the constant hazard ratio between the treatment arm and the control arm, that is

$$\lambda_1(t) = e^{\beta_0}\lambda_0(t)$$

and the difference of the odds ratio of cure rates between two arms is a constant, which can be written as $\text{logit}(\pi_1) = \text{logit}(\pi_0) + \gamma_0$, where β_0 and γ_0 are unknown parameters. When $\pi_0 = \pi_1 = 0$, it reduces to the standard PH model.

For a clinical trial with a proportion of patients being cured, we are interested in testing

$$H_0 : S_1(t) = S_0(t) \text{ and } \pi_1 = \pi_0,$$

which is equivalent to

$$H_0 : \beta_0 = \gamma_0 = 0$$

. Based on the alternative hypotheses ($H_{a,n}$: $\beta_0 = \beta_a/\sqrt{n}, \gamma_0 = \gamma_a/\sqrt{n}$) and the log-rank test, Wang et al. [29] has shown that to achieve a power of $1-\theta$, the total sample size for the PH mixture cure model can be determined by

$$n = \frac{(Z_\theta + Z_{\alpha/2})^2 \int_0^\infty S_C(t) f_0(t) dt}{p(1-p)\beta_0^2(1-\pi_0)\left\{ \int_0^\infty m(\gamma_0, \beta_0, \pi_0) S_C(t) f_0(t) dt \right\}^2}, \tag{3.3}$$

where $m(\gamma_0, \beta_0, \pi_0) = \pi_0\{\gamma_0/\beta_0 + \Lambda_0(t)\}/S_0^*(t) - 1$. When $\pi_0 = 0$, $m(\gamma_0, \beta_0, \pi_0) = -1$. The above sample size formula is reduced to the standard PH model sample size formula as given in (3.1).

Let τ_a denote the accrual period, τ_f denote the follow-up time and τ denote the total study length with $\tau = \tau_a + \tau_f$. Let $g(t)$ denote the probability density function of accrual times and three (uniform, increasing and decreasing) accrual patterns are considered in the package. We also assume that the only censoring is due to administrative censoring at time τ, and there is no loss to follow-up or competing risks. The probability density functions $g(t)$ of accrual times and their corresponding survival functions $S_C(t)$ of the censoring times for the uniform, increasing and decreasing accruals are summarized in Table 3.1.

Table 3.1 Density functions $g(t)$ of accrual times and the corresponding survival functions $S_C(t)$ of censoring times.

Accrual	$g(t)$	$S_C(t)$
Uniform	$g(t) = \begin{cases} \frac{1}{\tau_a} & \text{if } 0 < t \le \tau_a \\ 0 & \text{otherwise} \end{cases}$	$S_C(t) = \begin{cases} 1 & \text{if } t \le \tau_f \\ \frac{\tau_a + \tau_f - t}{\tau_a} & \text{if } \tau_f < t \le \tau_a + \tau_f \\ 0 & \text{if } t > \tau_a + \tau_f \end{cases}$
Increasing	$g(t) = \begin{cases} \frac{2t}{\tau_a^2} & \text{if } 0 < t \le \tau_a \\ 0 & \text{otherwise} \end{cases}$	$S_C(t) = \begin{cases} 1 & \text{if } t \le \tau_f \\ \frac{(\tau_a + \tau_f - t)^2}{\tau_a^2} & \text{if } \tau_f < t \le \tau_a + \tau_f \\ 0 & \text{if } t > \tau_a + \tau_f \end{cases}$
Decreasing	$g(t) = \begin{cases} \frac{2(\tau_a - t)}{\tau_a^2} & \text{if } 0 < t \le \tau_a \\ 0 & \text{otherwise} \end{cases}$	$S_C(t) = \begin{cases} 1 & \text{if } t \le \tau_f \\ 1 - \frac{(\tau_f - t)^2}{\tau_a^2} & \text{if } \tau_f < t \le \tau_a + \tau_f \\ 0 & \text{if } t > \tau_a + \tau_f \end{cases}$

Examples Under Parametric Assumption

Uniform Accrual and Exponential Distribution

The uniform accrual assumes that patients enter a study at a constant rate $1/\tau_a$. The exponential distribution with the rate of λ_0 assumes that the patients in the control arm has mean survival time of $1/\lambda_0$ and hazard risk in the control arm is changed by a constant λ_0, that is $\lambda_0(t) = \lambda_0$, $\Lambda_0(t) = \lambda_0 t$ and $S_0(t) = e^{-\lambda_0 t}$. Plugging the defined survival functions $S_C(t)$ and other information into formula (3.3), the sample size is calculated as

$$n = \frac{(Z_\theta + Z_{\alpha/2})^2 \left(\int_0^{\tau_f} \lambda_0 e^{-\lambda_0 t} dt + \int_{\tau_f}^{\tau_a+\tau_f} \frac{\tau_a+\tau_f-t}{\tau_a} \lambda_0 e^{-\lambda_0 t} dt \right)}{p(1-p)\beta_0^2(1-\pi_0) \left\{ \int_0^{\tau_f} m(\gamma_0, \beta_0, \pi_0)\lambda_0 e^{-\lambda_0 t} dt + \int_{\tau_f}^{\tau_a+\tau_f} \frac{\tau_a+\tau_f-t}{\tau_a} m(\gamma_0, \beta_0, \pi_0)\lambda_0 e^{-\lambda_0 t} dt \right\}^2}, \tag{3.4}$$

where $m(\gamma_0, \beta_0, \pi_0) = \frac{\pi_0(\gamma_0/\beta_0 + \lambda_0 t)}{\pi_0 + (1-\pi_0)e^{-\lambda_0 t}} - 1$.

Increasing Accrual and Weibull Distribution

The increasing accrual assumes that patients enter a study with the density function of $g(t) = \frac{2t}{\tau_a^2}$. The Weibull distribution with scale parameter λ_0 and shape parameter k is assumed for survival times of uncured patients, which can be written as Comparing to the exponential assumption, the Weibull distribution allows increasing hazard rate ($k > 1$), constant hazard rate ($k = 1$) and decreasing hazard rate ($0 < k < 1$). The sample size is calculated as

$$n = \frac{(Z_\theta + Z_{\alpha/2})^2 \left(\int_0^{\tau_f} \lambda_0(t) S_0(t) dt + \int_{\tau_f}^{\tau_a+\tau_f} \frac{(\tau_a+\tau_f-t)^2}{(\tau_a)^2} \lambda_0(t) S_0(t) dt \right)}{p(1-p)\beta_0^2(1-\pi_0) \left\{ \int_0^{\tau_f} m(\gamma_0, \beta_0, \pi_0)\lambda_0(t) S_0(t) dt + \int_{\tau_f}^{\tau_a+\tau_f} \frac{(\tau_a+\tau_f-t)^2}{(\tau_a)^2} m(\gamma_0, \beta_0, \pi_0)\lambda_0(t) S_0(t) dt \right\}^2}, \tag{3.5}$$

where $m(\gamma_0, \beta_0, \pi_0) = \frac{\pi_0(\gamma_0/\beta_0+\lambda_0 t^k)}{\pi_0+(1-\pi_0)e^{-\lambda_0 t^k}} - 1$, $\lambda_0(t) = \lambda_0 k(\lambda_0 t)^{k-1}$, $\Lambda_0(t) = (\lambda_0 t)^k$ and $S_0(t) = e^{-(\lambda_0 t)^k}$.

Example Under Nonparametric Estimation of Parameters $(\hat{S}_0(t),\ \hat{\pi}_0,\ \hat{\gamma}_0,\ \hat{\beta}_0)$

Let $t_{(1)} < t_{(2)} < \cdots < t_{(k)}$ be the distinct failure times. If observed/historical data is available, the survival function $S_0(t)$, cure rate π_0, log odds ratio γ_0 and log hazard ratio β_0 can be estimated from the PH mixture cure model, which is implemented by `smcure` package in R [4]. In this situation, only α, θ, p and accrual pattern need to be specified. The sample size formula for nonparametric estimation is written as

$$n = \frac{(Z_\theta + Z_{\alpha/2})^2 \sum_{i=1}^{k} \hat{S}_0(t_{(i)}) S_C(t_{(i)})}{p(1-p)\beta_0^2(1-\pi_0)\Big\{\sum_{i=1}^{k} \hat{S}_0(t_{(i)}) S_C(t_{(i)}) \hat{m}(\gamma_0, \beta_0, \pi_0; t_{(i)})\Big\}^2}, \tag{3.6}$$

where $\hat{m}(\gamma_0, \beta_0, \pi_0; t_i) = \pi_0\{\gamma_0/\beta_0 + \hat{\Lambda}_0(t)\}/\hat{S}_0^*(t) - 1$, $\hat{\Lambda}_0(t) = \log(-\hat{S}_0(t))$, $\hat{S}_0^*(t) = \pi_0 + (1-\pi_0)\hat{S}_0(t)$.

3.4 Package Description

The sample size formula (3.3) under the exponential or the Weibull distribution with different accrual patterns and formula (3.6) are implemented in the `NPHMC` package. The `NPHMC` function in the package can be called with the following syntax:

```
NPHMC <- function(power, alpha, accrualtime, followuptime, p,
      accrualdist=c("uniform","increasing","decreasing"), hazardratio,
      oddsratio, pi0, survdist=c("exp","weib"), k, lambda0, data=NULL)
```

The arguments are:

- `power`: specifies the required power. The default power = 80%.
- `alpha`: specifies the level of significance of the statistical test. The default is 0.05.
- `accrualtime`: specifies the length of accrual period.
- `followuptime`: specifies the follow-up time.
- `p`: specifies the proportion of subjects in each arm. The default p = 0.5.
- `accrualdist`: specifies the accrual rate distribution. It can be `"uniform"`, `"increasing"` or `"decreasing"`.
- `hazardratio`: specifies the hazard ratio of uncured patients between two arms, which is equivalent to $e^{\beta_0} = \lambda_1(t)/\lambda_0(t)$. The value must be greater than 0.
- `oddsratio`: specifies the odds ratio of cure rates between two arms, which is equivalent to $e^{\gamma_0} = \frac{\pi_1}{1-\pi_1}/\frac{\pi_0}{1-\pi_0}$. The value should be greater than 0 if cure rates exist. It can be 0 if there is no cure rate.
- `pi0`: specifies the cure rate for the control arm, which is between 0 and 1.
- `survdist`: specifies the survival distribution of uncured patients. It can be `"exp"` or `"weib"`.
- `k`: if `survdist` = `"weib"`, the shape parameter `k` needs to be specified. By default `k` = 1, which refers to the exponential distribution.
- `lambda0`: specifies the scale parameter of exponential distribution or Weibull distribution for survival times of uncured patients in control arm.

  The density function of Weibull distribution with shape parameter k and scale parameter λ_0 is given by

$$f(t) = \lambda_0 k(\lambda_0 t)^{k-1} \exp(-(\lambda_0 t)^k),\ for\ t > 0$$

  .

- `data`: if observed/historical data is available, the sample size can be calculated based on the nonparametric estimators from the PH mixture cure model by `smcure` package in R. The data must contain three columns with the order of `"Time"`, `"Status"` and `"X"` where `"Time"` refers to time to event of interest, `"Status"` refers to censoring indicator (1 = event of interest happens, and 0 = censoring) and `"X"` refers to arm indicator (1 = treatment and 0 = control). By default, `data = NULL`.

Output:

If `data` = NULL, the output will display

- `PH Mixture Cure Model: n`
- `Standard PH Model: n`

When `data` is specified, the output will first display the estimators from the `smcure` package in R, and then show results from the `NPHMC` package.

- `Estimators from smcure package`
- `PH Mixture Cure Model: n`
- `Standard PH Model: n`

3.5 Simulation Study

In this section, we conduct a simulation study to investigate the performance of the `NPHMC` package based on the PH mixture cure model. Two sets of results are reported. One is based on the fully parametric approach, and the other is based on the nonparametric approach.

The following settings are used in the simulation study: (1) an exponential distribution with parameter $\lambda_0 = 1$, and a Weibull distribution with shape parameter $k = 2$ and scale parameter $\lambda_0 = 1$ are assumed for survival distributions of uncured

patients; (2) an accrual period of 3 years and a follow-up time of 4 years; (3) an equal allocation $p = 0.5$; (4) a number of 500 observations is generated in each dataset; (5) simulation results are based on 200 replications.

Table 3.2 Comparison of Exponential Parametric Sample Size Estimation with Nonparametric Sample Size Estimation (200 replications)

π_0	π_1	OR	λ_0	λ_1	HR	k	Accrual Rate	Parametric Sample Size	Nonparametric Sample Size
0.2	0.4	2.667	1	1/2	0.5	1	Uniform	110	111
							Increasing	108	112
							decreasing	112	112
	0.45	3.273					Uniform	88	89
							Increasing	87	90
							decreasing	89	90
	0.5	4.000					Uniform	73	74
							Increasing	72	74
							decreasing	73	74
0.2	0.5	4.000	1	1/2	0.5	1	Uniform	73	74
							Increasing	72	74
							decreasing	73	74
				1/2.5	0.4		Uniform	59	60
							Increasing	58	60
							decreasing	59	60
				1/3	0.3		Uniform	50	51
							Increasing	49	51
							decreasing	51	51

We first compare the parametric estimation approach based on the exponential distribution with the nonparametric estimation approach in Table 3.2. We fix $\pi_0 = 0.2$, $\lambda_0 = 1$, and then set $\pi_1 = (0.4, 0.45, 0.5)$ and $\lambda_1 = (1/2, 1/2.5, 1/3)$ respectively, which correspond to the values of `oddsratio` = (2.6667, 3.2727, 4) and `hazardratio` = (0.5, 0.4, 0.3). In Table 3.3, we consider the Weibull distribution with $k = 2$. The same settings of odds ratio and hazards ratio are used. Both tables show that the

Table 3.3 Comparison of Weibull Parametric Sample Size Estimation with Nonparametric Sample Size Estimation (200 replications)

π_0	π_1	OR	λ_0	λ_1	HR	k	Accrual Rate	Parametric Sample Size	Nonparametric Sample Size
0.2	0.4	2.667	1	0.707	0.5	2	Uniform	115	125
							Increasing	115	124
							decreasing	115	127
	0.45	3.272					Uniform	92	96
							Increasing	92	93
							decreasing	92	97
	0.5	4.000					Uniform	75	77
							Increasing	75	76
							decreasing	75	79
0.2	0.5	4.000	1	0.707	0.5	2	Uniform	75	77
							Increasing	75	76
							decreasing	75	79
				0.632	0.4		Uniform	61	64
							Increasing	61	61
							decreasing	61	64
				0.548	0.3		Uniform	48	50
							Increasing	48	49
							decreasing	48	50

results from the nonparametric sample size estimation are quite close to those based on the parametric approach.

3.6 Examples

Parametric Sample Size Estimation

If the survival curve in each arm is assumed to follow exponential distribution or Weibull distribution, besides the specifications of `power`, `alpha`, `accrualtime`, `follow uptime` and `p`, the user needs to give `accrualdist`, `survdist`, `k`, `lambda0`, and assumption of relationship between two arms, such as `hazardratio` and `oddsratio` in

order to calculate sample size for the PH mixture cure model.

For example, a survival trial will follow a uniform accrual with an accrual period of 3 years and a follow-up period of 4 years with equal amount of patients in each arm (p = 0.5). The mean life of uncured patients in control arm will be 2 years and the mean life of uncured patients in treatment arm will be 2.5 years. Assume both arms follow the exponential distribution. Further, assume cure rates are $\pi_0 = 0.1$ and $\pi_1 = 0.2$ for the control arm and treatment arm. At 5% significance level, to detect a 25% improvement in mean survival time from 2 to 2.5 years and achieve 90% power of statistical test, the estimated sample size can be obtained by the following code:

```
> NPHMC(power=0.90,alpha=0.05,accrualtime=3,followuptime=4,p=0.5,
accrualdist="uniform",hazardratio=2/2.5,oddsratio=2.25,pi0=0.1,
survdist="exp",k=1,lambda0=0.5)
```

The output is:

```
=====================================================================
SAMPLE SIZE CALCULATION FOR PH MIXTURE CURE MODEL AND STANDARD PH MODEL
=====================================================================
PH Mixture Cure Model: n = 429
Standard PH Model: n = 908
```

A sample size of 429 patients will be needed to achieve a power of 90% based on the PH mixture cure model. The sample size from the standard PH model is 908 which is overestimated if there exits a cure rate.

Nonparametric Sample Size Estimation When Observed/Historical Data is Available

We illustrate the application of `NPHMC` package by melanoma data from the ECOG phase III clinical trial e1684 [16]. The ECOG trial e1684 was a two-arm phase III

clinical trial comparing high dose interferon alpha-2b with an observation arm. The primary endpoint was relapse-free survival (RFS), with RFS defined as the time from randomization until progression of the tumor or death. Note that our intention here is not to re-design the trial but to show the application of the package.

If an observed/historical data is given, users only need to specify `power`, `alpha`, `accrualtime`, `followuptime`, `p`, `accrualdist` and `data`. The hazard ratio and cure rates can be directly estimated from the available data, therefore the sample size can be obtained by the following code:

```
> NPHMC(power=0.80,alpha=0.05,accrualtime=4,followuptime=3,p=0.5,
  accrualdist="uniform",data=e1684szdata)
```

The output is:

```
Call:
smcure(formula = Surv(Time, Status) ~ X, cureform = ~Z, data = data,
    model = "ph", Var = FALSE)


Cure probability model:
              Estimate
(Intercept)  1.2850677
Z           -0.5455204



Failure time distribution model:
    Estimate
X -0.1643542

======================================================================
SAMPLE SIZE CALCULATION FOR PH MIXTURE CURE MODEL AND STANDARD PH MODEL
```

```
======================================================================
PH Mixture Cure Model with KM estimators: n = 454
Standard PH Model with KM estimators: n = 251
```

The package first fitted the data using the `smcure` R package with the treatment as a covariate. The log hazard ratio is estimated as $\hat{\beta}_0$ = -0.164. The coefficients of logistic regression model for cure probability model are 1.285 and -0.5455, which lead to cure rates for the observation arm and the interferon arm as $\hat{\pi}_0 = 1 - \frac{e^{1.285}}{1+e^{1.285}} = 0.2167$ and $\hat{\pi}_1 = 1 - \frac{e^{1.285-0.5455}}{1+e^{1.285-0.5455}} = 0.3231$. To achieve a power of 80%, a sample size of 454 is required based on the estimates from the PH mixture cure model. A sample size of 251 is calculated based on the standard PH model assumption, which will lead to a underpowered trial if there is a cure fraction.

3.7 Conclusions

We develop an R package to estimate the sample size of the PH mixture cure model. Comparing to existing software, the main advantage of this package is to allow a cure fraction in survival trial. Besides that, this package can allow patients to enter study with different patterns and also different hazard patterns for the uncured patients. Therefore, the `NPHMC` package provides an important and flexible tool in sample size design in survival trial with or without cure fractions.

3.8 Availability

The package `NPHMC` and the relevant documentation can be freely downloaded from CRAN webpage http://cran.r-project.org/package=NPHMC.

Chapter 4

New Estimation Method for Semiparametric PH Mixture Cure Model with Competing Risks Data

4.1 Introduction

Competing risks data are commonly seen in medical research particularly in survival analysis when subjects are at risk of failure from K different causes. When one event occurs, it precludes the occurrence of any other events. The PH mixture cure model is commonly used regression model that accounts for the cure fraction. Considering two types of failures $k = 1, 2$, within the mixture cure model framework, it is assumed that an individual will fail from the event of interest or other risks. The following flow chart can show you how cure fraction works in competing risks data framework.

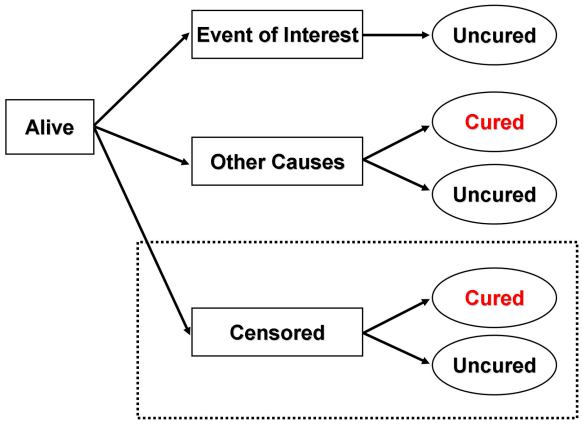

Figure 4.1 Cure Model with Competing Risks Data.

In Figure 4.1, it is obvious that cure fraction not only exits in censored patients but also in those patients who died of other causes. Ignoring the competing risks would lead the bias in estimating cure rates. In this project, I propose a new estimation method for semiparametric PH mixture cure model with competing risks data. The estimation is based on maximum likelihood of the full likelihood. In next sections, I will discuss data and model, computational method, simulation study and real data analysis for the semiparametric PH mixture cure model with competing risks data.

4.2 Data and Model

Let T_1 be the failure time from the event of interest, T_2 be the failure time from all other risks, T_i be the event time where $T = (T_1 \cap T_2)$ and C_i be the right censoring times for ith individual. $\varepsilon_i \in (1, \ldots, K)$ indicates the cause of failure. We consider $K = 2$ in this study ($\varepsilon = 1$ for event of interest; $\varepsilon = 2$ for other causes). Let $\tilde{T} = min(T, C)$ and $\delta = I(T \leq C)$. $\mathbf{z}$ be a vector of covariates.

Let $\mathbf{O} = (\tilde{T}_i, \delta_i, \delta_i\varepsilon_i, \mathbf{z_i})$ denote the observed competing risks data for the ith individual $i = 1, \cdots, n$, where $\tilde{T}_i$ is the observed survival time, δ_i is the censoring indicator with $\delta_i = 1$ for the uncensored time and $\delta_i = 0$ for the censored time, and $\mathbf{z_i}$ are the possible covariates that affect the cure probability, cause of death probability and marginal survival probability for specific cause of death, respectively. We assume that (T_i, ε_i) are independent of C_i given covariates.

Let Y be the indicator that an individual will eventually ($Y = 1$) or never ($Y = 0$) experience the event of interest, with the probability of $\pi(\mathbf{bz})$. Usually, a logit link function

$$\pi(\mathbf{bz}) = \frac{\exp(\mathbf{bz})}{1 + \exp(\mathbf{bz})},$$

where $\mathbf{b}$ is a vector of unknown parameters, is used to model the probability of being uncure. The probability of being cured is $1 - \pi(\mathbf{bz})$.

Let $\pi(\boldsymbol{\theta}\mathbf{z})$ be the probability of failure from the event of interest and $1-\pi(\boldsymbol{\theta}\mathbf{z})$ be the probability of failure from other risks. It is obvious that if an individual is cured $(Y=0)$, then the individual must fail from other risks $(\varepsilon=2)$. Given an individual is uncured $(Y=1)$, the conditional probability of failure from the event of interest is assumed to have the following logistic form,

$$P(\varepsilon=1|Y=1)=\frac{\exp(\theta\mathbf{z})}{1+\exp(\theta\mathbf{z})}.$$

Therefore, the probability of failure from the jth cause is given by

$$P(\varepsilon=j)=\begin{cases}\pi(\mathbf{bz})\pi(\boldsymbol{\theta}\mathbf{z}) & \text{if } j=1\\ 1-\pi(\mathbf{bz})+\pi(\mathbf{bz})[1-\pi(\boldsymbol{\theta}\mathbf{z})] & \text{if } j=2\end{cases}$$

where $P(\varepsilon=1)+P(\varepsilon=2)=1$.

Let $S_j(t;\mathbf{z})$ be the conditional survival function given that failure is due to the jth cause. The survival function $S_j(t;\mathbf{z})$ is given by

$$S_j(t;\mathbf{z})=P(T_i>t|\varepsilon_i=j)=\begin{cases}S_{0j}(t)^{\exp(\boldsymbol{\beta_j}\mathbf{z})} & \text{if } j=1\\ S_{0j}(t)^{\exp(\boldsymbol{\beta_j}\mathbf{z})} & \text{if } j=2\end{cases}$$

where $S_{0j}(t)$ is a cause-specific baseline survival function and $\boldsymbol{\beta_j}$ is a vector of regression coefficients for jth cause.

4.3 Computational Method

Let $\boldsymbol{\Theta}=(\mathbf{b},\boldsymbol{\theta},\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,S_{01}(t),S_{02}(t))$ be the unknown parameters. The EM algorithm is used to estimate the parameters of interest in the mixture cure model with competing risks data. The estimation is based on the maximum likelihood method.

LIKELIHOOD

On the basis of the observed data $\mathbf{O} = (\tilde{T}_i, \delta_i, \delta_i\varepsilon_i, \mathbf{z_i})$, the observed likelihood function for the unknown parameters $\boldsymbol{\Theta}$ under the PH mixture cure model is given by

$$\begin{aligned} L_o = \prod_{i=1}^{n} & \{\pi(\mathbf{bz_i})\pi(\boldsymbol{\theta}\mathbf{z_i})h_1(t_i)S_1(t_i)\}^{I(\varepsilon_i=1)\delta_i} \\ & \{[(1-\pi(\mathbf{bz_i})) + \pi(\mathbf{bz_i})(1-\pi(\boldsymbol{\theta}\mathbf{z_i}))]h_2(t_i)S_2(t_i)\}^{I(\varepsilon_i=2)\delta_i} \\ & \{\pi(\mathbf{bz_i})\pi(\boldsymbol{\theta}\mathbf{z_i})S_1(t_i) + [(1-\pi(\mathbf{bz_i})) + \pi(\mathbf{bz_i})(1-\pi(\boldsymbol{\theta}\mathbf{z_i}))]S_2(t_i))\}^{1-\delta_i} \end{aligned} \tag{4.1}$$

where $\pi(\mathbf{bz_i})$ is probability of being uncured for ith individual, $\pi(\boldsymbol{\theta}\mathbf{z_i})$ is probability of failing from event of interest for ith individual, $S_1(t_i)$ is the survival function from event of interest and $S_2(t_i)$ is the survival function due to other causes for ith individual. $h_j(\cdot)$ is the hazard function corresponding to $S_j(\cdot)$ for jth type of risk, $j = 1, 2$.

Given the cure indicator $\mathbf{y} = (y_1, y_2, \cdots, y_n)$ and the observed data $\mathbf{O}$, the complete likelihood function can be expressed as

$$\begin{aligned} L_c = \prod_{i=1}^{n} & \{\pi(\mathbf{bz_i})\pi(\boldsymbol{\theta}\mathbf{z_i})h_1(t_i)S_1(t_i)\}^{I(\varepsilon_i=1)\delta_i} \\ & \{[1-\pi(\mathbf{bz_i})]^{1-y_i}[\pi(\mathbf{bz_i})(1-\pi(\boldsymbol{\theta}\mathbf{z_i}))]^{y}h_2(t_i)S_2(t_i)\}^{I(\varepsilon_i=2)\delta_i} \\ & \{\pi(\mathbf{bz_i})\pi(\boldsymbol{\theta}\mathbf{z_i})S_1(t_i)\}^{I(\varepsilon_i=1)(1-\delta_i)} \\ & \{[1-\pi(\mathbf{bz_i})])^{1-y_i}[\pi(\mathbf{bz_i})(1-\pi(\boldsymbol{\theta}\mathbf{z_i}))]^{y_i}S_2(t_i))\}^{I(\varepsilon_i=2)(1-\delta_i)} \end{aligned} \tag{4.2}$$

The logarithm of the complete likelihood function can be written as

$l_c(\mathbf{b}, \boldsymbol{\theta}, \boldsymbol{\beta_1}, \boldsymbol{\beta_2}; \mathbf{O}, \mathbf{y}) = l_{c_1}(\mathbf{b}; \mathbf{O}, \mathbf{y}) + l_{c_2}(\boldsymbol{\theta}; \mathbf{O}, \mathbf{y}) + l_{c_3}(\boldsymbol{\beta_1}; \mathbf{O}, \mathbf{y}) + l_{c_4}(\boldsymbol{\beta_2}; \mathbf{O}, \mathbf{y})$, where

$$l_{c1}(\mathbf{b}) = \sum_{i=1}^{n}[I(\varepsilon_i = 1) + I(\varepsilon_i = 2)y_i]\log\pi(\mathbf{bz_i}) + \sum_{i=1}^{n} I(\varepsilon_i = 2)(1-y_i)\log(1-\pi(\mathbf{bz_i})) \tag{4.3}$$

$$l_{c2}(\boldsymbol{\theta}) = \sum_{i=1}^{n} I(\varepsilon_i = 1) \log \pi(\theta \mathbf{z_i}) + \sum_{i=1}^{n} I(\varepsilon_i = 2) y_i \log(1 - \pi(\theta \mathbf{z_i})) \tag{4.4}$$

$$l_{c3}(\boldsymbol{\beta_1}, S_{01}(t)) = \sum_{i=1}^{n} [I(\varepsilon_i = 1)\delta_i \log h_1(t_i) + I(\varepsilon_i = 1) \log S_1(t_i) \tag{4.5}$$

$$l_{c4}(\boldsymbol{\beta_2}, S_{02}(t)) = \sum_{i=1}^{n} [I(\varepsilon_i = 2)\delta_i \log h_2(t_i) + I(\varepsilon_i = 1) \log S_2(t_i) \tag{4.6}$$

with respect to the unknown parameters $\boldsymbol{\Theta}^{(\mathbf{m})} = (\mathbf{b}^{(m)}, \boldsymbol{\theta}^{(m)}, \boldsymbol{\beta}_1^{(m)}, \boldsymbol{\beta}_2^{(m)}, S_{01}(t)^{(m)}, S_{02}(t)^{(m)})$.

EM Algorithm

The E-step in the EM algorithm computes the conditional expectation of the complete log-likelihood with respect to the three unobserved probabilities $P(Y = 1, \varepsilon = 1)$, $P(Y = 1, \varepsilon = 2)$, and $P(Y = 0, \varepsilon = 2)$. These three probabilities sum to 1 and can be given by

$$P(Y = 1, \varepsilon = 1) = \delta I(\varepsilon = 1) + (1 - \delta) \frac{\pi(\mathbf{b})\pi(\boldsymbol{\theta}) S_1(t)}{[1 - \pi(\mathbf{b})] S_2(t) + \pi(\mathbf{b})\pi(\boldsymbol{\theta}) S_1(t) + \pi(\mathbf{b})[1 - \pi(\boldsymbol{\theta})] S_2(t)} \tag{4.7}$$

$$P(Y = 1, \varepsilon = 2) = \delta I(\varepsilon = 2) \frac{\pi(\boldsymbol{b})[1 - \pi(\boldsymbol{\theta})] S_2(t)}{[1 - \pi(\boldsymbol{b})] S_2(t) + \pi(\boldsymbol{b})[1 - \pi(\boldsymbol{\theta})] S_2(t)} + (1 - \delta) \frac{\pi(\boldsymbol{b})[1 - \pi(\boldsymbol{\theta})] S_2(t)}{[1 - \pi(\boldsymbol{b})] S_2(t) + \pi(\boldsymbol{b})\pi(\boldsymbol{\theta}) S_1(t) + \pi(\boldsymbol{b})[1 - \pi(\boldsymbol{\theta})] S_2(t)} \tag{4.8}$$

$$P(Y = 0, \varepsilon = 2) = \delta I(\varepsilon = 2) \frac{[1 - \pi(\boldsymbol{b})] S_2(t)}{[1 - \pi(\boldsymbol{b})] S_2(t) + \pi(\boldsymbol{b})[1 - \pi(\boldsymbol{\theta})] S_2(t)} + (1 - \delta_i) \frac{[1 - \pi(\boldsymbol{b})] S_2(t)}{[1 - \pi(\boldsymbol{b})] S_2(t) + \pi(\boldsymbol{b})\pi(\boldsymbol{\theta}) S_1(t) + \pi(\boldsymbol{b})[1 - \pi(\boldsymbol{\theta})] S_2(t)} \tag{4.9}$$

Let $\hat{p}_{11,i}^{(m)} = P(y_i = 1, \varepsilon_i = 1|\mathbf{O}, \mathbf{\Theta^{(m)}})$ and $\hat{p}_{\varepsilon,1i}^{(m)} = P(\varepsilon_i = 1|\mathbf{O}, \mathbf{\Theta^{(m)}})$, then $\hat{p}_{\varepsilon,1i}^{(m)} = \hat{p}_{11,i}^{(m)}$. Let $\hat{p}_{12,i}^{(m)} = P(y_i = 1, \varepsilon_i = 2|\mathbf{O}, \mathbf{\Theta^{(m)}})$ and $\hat{p}_{02,i}^{(m)} = P(y_i = 0, \varepsilon_i = 2|\mathbf{O}, \mathbf{\Theta^{(m)}})$ and $\hat{p}_{\varepsilon,2i}^{(m)} = P(\varepsilon_i = 2|\mathbf{O}, \mathbf{\Theta^{(m)}})$, then $\hat{p}_{\varepsilon,2i}^{(m)} = \hat{p}_{12,i}^{(m)} + \hat{p}_{02,i}^{(m)}$. The expectations of (4.3),(4.4),(4.5) and (4.6) can be written as

$$E(l_{c_1}) = \sum_{i=1}^{n} \left([\hat{p}_{11,i}^{(m)} + \hat{p}_{12,i}^{(m)}] \log[\pi(\mathbf{bz_i})] + \hat{p}_{02,i}^{(m)} \log[1 - \pi(\mathbf{bz_i})] \right), \tag{4.10}$$

$$E(l_{c_2}) = \sum_{i=1}^{n} \left(\hat{p}_{11,i}^{(m)} \log[\pi(\boldsymbol{\theta}\mathbf{z_i})] + \hat{p}_{12,i}^{(m)} \log[1 - \pi(\boldsymbol{\theta}\mathbf{z_i})] \right), \tag{4.11}$$

$$E(l_{c_3}) = \sum_{i=1}^{n} \left(\delta_i \hat{p}_{\varepsilon,1i}^{(m)} \log h_1(t; \boldsymbol{\beta_1}) + \hat{p}_{\varepsilon,1i}^{(m)} \log S_1(t; \boldsymbol{\beta_1}) \right), \tag{4.12}$$

$$E(l_{c_4}) = \sum_{i=1}^{n} \left(\delta_i \hat{p}_{\varepsilon,2i}^{(m)} \log h_2(t; \boldsymbol{\beta_2}) + \hat{p}_{\varepsilon,2i}^{(m)} \log S_2(t; \boldsymbol{\beta_2}) \right). \tag{4.13}$$

The M-step in the EM algorithm is to maximize (4.10), (4.11), (4.12) and (4.13) with respect to the unknown parameters $\mathbf{\Theta} = (\mathbf{b}, \boldsymbol{\theta}, \boldsymbol{\beta_1}, \boldsymbol{\beta_2}, S_{01}(t), S_{02}(t))$. The parameters in equation (4.10) and equation (4.11) can be easily estimated by '`optim`' function in R.

Peng and Dear [22] and Sy and Taylor [26] proposed a partial likelihood type method to estimate $\boldsymbol{\beta_1}$ and $\boldsymbol{\beta_2}$ without specifying the baseline hazard functions. The estimating equation (4.12) and (4.13) can be rewritten as

$$\log \prod_{i=1}^{n} [h_0(t_i) \exp(\boldsymbol{\beta}\mathbf{z_i} + \log(p_{\varepsilon,ji}^{(m)}))]^{\delta_i} S_0(t_i)^{\exp(\boldsymbol{\beta}\mathbf{z_i} + \log(p_{\varepsilon,ji}^{(m)}))} \tag{4.14}$$

which is similar to the log-likelihood function of the standard PH model with the additional offset variable $\log(p_{\varepsilon,ji}^{(m)}), j = 1, 2$. Therefore, the parameters in equation (4.12) and (4.13) can be estimated by '`coxph`' package in R. A detailed presentation can be found in Peng [20], Peng and Dear [22], and Sy and Taylor [26].

Estimation of Baseline Survival Function

In order to proceed the E-step in the EM algorithm, we need to update the estimated survival function $S_1(t)$ and $S_2(t)$ at each iteration. Let $t_{(j1)} < t_{(j2)} < \cdots < t_{(jk)}$ be the distinct uncensored failure times due to the jth cause (j=1, 2), $d_{t_{(jk)}}$ denote the number of failures due to cause j at time $t_{(jk)}$ and $R(t_{(jk)})$ denote the risk set at time $t_{(jk)}$. The nonparametric Breslow-type estimator for baseline survival function $S_{0j}(t)$ is given by

$$\hat{S}_{0j}(t) = \exp\left(-\sum_{k:t_{(jk)}\leq t} \frac{d_{t_{(jk)}}}{\sum_{i\in R(t_{(jk)})} p_{\varepsilon,ji}^{(m)} e^{\beta \mathbf{z_i}}}\right). \tag{4.15}$$

4.4 Simulation

In this section, we consider the sample sizes of $n = 200$, $n = 500$ and $n = 800$ respectively. Two distinct causes of failure is assumed in this study. The covariate $\mathbf{z}$ is a binary variable, which is generated independently from the binary distribution with probability of 0.5. We assume $\mathbf{b} = (2, -1)$, $\boldsymbol{\theta} = (-1, 0.5)$, $\boldsymbol{\beta_1} = -0.5$ and $\boldsymbol{\beta_2} = -1$. The cure indicator Y is generated from the binary distribution with probability of $\pi(\mathbf{bz})$. The censoring time is generated from a uniform distribution $U(c_1, c_2)$, where c_1 and c_2 are some constants to obtain the desired censoring rate. If the ith failure time is greater than the ith censoring time, it is taken to be censored at this censoring time. In the simulation, we consider two different sets of values for c_1 and c_2 so that different censoring scenarios can be investigated. The survival times of uncure patients are generated from a Weibull distribution where $S_j(t) = \exp[-(\lambda_j t)^k \exp(\boldsymbol{\beta_j}\mathbf{z})], j = 1, 2$ ($\lambda_1 = 0.5$, $\lambda_2 = 1$ and $k = 1$) and a Lognormal distribution log N(0,1). The results for different censoring scenarios are presented in Table 3.2. For each simulation set, we generated 500 independent samples.

Tables 4.1 and 4.2 present the estimated biases and MSE from the PH mixture cure model of six regression parameters b_0, b_1, θ_0, θ_1, β_0 and β_1 based on the logistic-

Weibull data and logistic-lognormal data.

Table 4.1 Estimates of parameters from Logistic-Weibull PH mixture cure model

Average censoring	Parameter	True value	Cure rate	n = 200		n = 500		n = 800	
				Bias	MSE	Bias	MSE	Bias	MSE
20.1	b_0	2	0.12	0.0243	0.1018	-0.0177	0.0346	-0.0212	0.0247
	b_1	-1	0.27	-0.0454	0.1430	-0.0396	0.0572	-0.0317	0.0388
	θ_0	-1		0.0089	0.0687	0.0086	0.0268	0.0063	0.0155
	θ_1	0.5		0.0847	0.1529	0.02132	0.0649	0.0196	0.0345
	β_0	-0.5		-0.0940	0.1681	-0.0365	0.0547	-0.0173	0.0301
	β_1	-1		0.0192	0.0485	0.0017	0.0162	0.0002	0.0104
33.5	b_0	1.3863	0.2	-0.5056	0.3281	-0.3758	1.7970	-0.5035	0.2956
	b_1	-1	0.4	-0.1124	0.1443	-0.1452	0.5357	-0.0689	0.0449
	θ_0	-1		0.2131	0.1136	0.2377	0.0820	0.2474	0.0773
	θ_1	0.5		0.2881	0.2347	0.2713	0.1393	0.2946	0.1257
	β_0	-0.5		0.0734	0.0630	0.0619	0.0275	0.0530	0.0160
	β_1	-1		-0.0169	0.0603	-0.0021	0.0220	0.0008	0.0151

Table 4.2 Estimates of parameters from Logistic-Lognormal PH mixture cure model

Average censoring	Parameter	True value	Cure rate	n = 200		n = 500		n = 800	
				Bias	MSE	Bias	MSE	Bias	MSE
23.3	b_0	2	0.12	0.0725	0.1092	0.0516	0.0421	0.0318	0.0225
	b_1	-1	0.27	-0.3256	0.2527	-0.3306	0.1633	-0.3299	0.1425
	θ_0	-1		-0.0420	0.0815	-0.0396	0.0329	-0.0543	0.0202
	θ_1	0.5		0.1084	0.1704	0.0330	0.0904	0.0317	0.0467
	β_0	-0.5		0.0247	0.2020	0.0707	0.0751	0.0981	0.0481
	β_1	-1		-0.0065	0.0497	-0.0086	0.0171	-0.0254	0.0112
34.5	b_0	1.3863	0.2	0.0096	0.0603	-0.0007	0.0259	-0.0137	0.0164
	b_1	-1	0.4	-0.2241	0.1576	-0.2313	0.0953	-0.2321	0.0784
	θ_0	-1		-0.1129	0.1053	0.1086	0.0474	-0.1185	0.0383
	θ_1	0.5		0.0507	0.2801	-0.0537	0.0955	-0.0501	0.0629
	β_0	-0.5		0.0674	0.3821	0.1498	0.1418	0.1646	0.0813
	β_1	-1		-0.0085	0.0606	-0.0165	0.0217	0.0252	0.0122

The simulation results show that point estimates do not depend on the assumption of the distribution. The bias of estimates based on PH mixture cure model are quite small on both Weibull distribution and lognormal distribution. The MSE of estimates get smaller as sample size increases. The same conclusion can be made when we increase cure rates from 12% to 20% in control group and 27% to 40% in treatment group.

4.5 Example

To illustrate the proposed estimation method for semiparametric PH mixture cure model with competing risks data, we will consider the prostate cancer clinical trial data. The survival times of 502 patients with prostate cancer entered the trial during 1967 to 1969 and were randomly allocated to different levels of treatment with the drug diethylstilbestrol (DES). These data have been analyzed and published by many authors [3, 14, 7]. Patients with missing covariates were excluded from this analysis. There were 483 patients with completion information on covariates.

In this analysis, we consider two types of failures: (1) death due to prostate cancer (2) death due to other causes. There were 125 patients who died from prostate cancer, and 219 patients who died from other causes. The remaining 139 survival times were censored. The censoring rate is 28.78%. Only covariate of drug treatment (0.0 or 0.2 mg coded as 0; 1.0 or 5.0 mg coded as 1) was considered in the analysis. The proposed semiparametric mixture cure model approach is adopted and the result is presented in Table 4.3.

Table 4.3 Maximum likelihood estimates for prostate cancer clinical trial data based on semiparametric mixture cure model

	Incidence Part		Latency Part	
Coefficient	Cure incidence	Cause of prostate cancer	Prostate cancer	Other causes
Constant	1.2713* (0.07)	-0.1943 (0.12)		
Treatment	-0.9129* (0.23)	-0.3011 (0.29)	0.0558 (0.23)	-0.6318* (0.24)

* p-value < 0.05

The data were fitted by PH mixture cure model based on completing risks data framework. The parameter $\boldsymbol{b}$ were estimated as $\boldsymbol{b} = (1.2713, -0.9129)$ which lead to cure rates for the drug treatment group (0.0 or 0.2 mg) and the drug treatment group (1.0 or 5.0 mg) as $\frac{e^{1.2713}}{1+e^{1.2713}} = 21.9\%$ and $\frac{e^{1.2713-0.9129}}{1+e^{1.2713-0.9129}} = 41.1\%$. Based on the results, we can conclude that treatment had significant effect on estimating cure rate

by logistic regression with p-value < 0.05. Treatment also had significant effect on estimating failure times due to other causes.

4.6 Conclusions and Discussion

We proposed an EM based algorithm for estimating PH mixture cure model with competing risks data. The estimation is based on maximum likelihood of the full likelihood, and estimation process allows the nonparametric maximum likelihood estimates of the baseline survival functions to be used in the estimation of the parameters.

Chapter 5

An Extension to Semiparametric AFT Mixture Cure Model and Its Application in R

Another important statistical model in medical research is the AFT model. If the AFT model is used to model the survival of uncure individuals, the cure model is called the AFT mixture cure model. In this Chapter, I will first introduce the standard AFT model and AFT mixture cure model, and then discuss the application of AFT mixture cure model in R.

5.1 Semiparametric AFT Mixture Cure Model

As mentioned before, the latency part of mixture cure model can be specified by either the PH model or the AFT model. The AFT model is specified as

$$\log(T) = \boldsymbol{\beta}\mathbf{x} + \varepsilon$$

where the distribution of the error term ε is unknown. The survival function can be written as

$$S(t|\mathbf{x}) = S_0(te^{\boldsymbol{\beta}\mathbf{x}}).$$

The mixture cure model is called the AFT mixture cure model if the assumption of latency part follows AFT model. This section will focus on an estimation method for the semiparametric AFT mixture cure model.

Let $f(\cdot)$ be the density probability function of ε and $S(\cdot)$ be the corresponding survival function. The conditional survival function of T, given that the patient is

not cured, is $S(\log(t) - \boldsymbol{\beta}\mathbf{x})$. Assuming the censoring is independent and noninformative, the contribution to the likelihood from the ith uncensored patient ($\delta_i = 1$) is $\pi(\mathbf{z_i})f(\log(t_i) - \boldsymbol{\beta}\mathbf{x})/t_i$ and $1 - \pi(\mathbf{z_i}) + \pi(\mathbf{z_i})S(\log(t_i) - \boldsymbol{\beta}\mathbf{x_i})$ from censored patient ($\delta_i = 0$).

Similar to the semiparametric PH mixture cure model, we define y_i as an indicator of cure status of the ith patient, namely, $y_i = 1$ if the patient is uncured and 0 otherwise, $i = 1, 2, \cdots, n$.. Given $\mathbf{y} = (y_1, y_2, \cdots, y_n)$ and $\mathbf{O}$, the complete data are available, The logarithm of the complete likelihood function can be written as

$l_c(\mathbf{b}, \boldsymbol{\beta}; \mathbf{O}, \mathbf{y}) = l_{c_1}(\mathbf{b}; \mathbf{O}, \mathbf{y}) + l_{c_2}(\boldsymbol{\beta}; \mathbf{O}, \mathbf{y})$, where

$$l_{c_1}(\mathbf{b}; \mathbf{O}, \mathbf{y}) = \sum_{i=1}^{n} y_i \log[\pi(\mathbf{z_i})] + (1 - y_i) \log[1 - \pi(\mathbf{z_i})], \tag{5.1}$$

$$l_{c_2}(\boldsymbol{\beta}; \mathbf{O}, \mathbf{y}) = \sum_{i=1}^{n} y_i \delta_i \log[h(\log(t_i) - \boldsymbol{\beta}\mathbf{x_i}), \mathbf{x_i})] + y_i \log[S(\log(t_i) - \boldsymbol{\beta}\mathbf{x_i})]. \tag{5.2}$$

and $h(\cdot) = f(\cdot)/S(\cdot)$ is the hazard function of ε. We can see that (5.1) and (5.2) are linear functions of latent variable Y. Therefore, we consider EM algorithm to estimate the unknown parameter $\boldsymbol{b}$ and $\boldsymbol{\beta}$.

The E-step in the EM algorithm computes the conditional expectation of the complete log-likelihood with respect to $y_i's$, given the observed data $\mathbf{O}$ and current estimates of parameters $\boldsymbol{\Theta}^{(\mathbf{m})} = (\mathbf{b}^{(m)}, \boldsymbol{\beta}^{(m)}, S_0^{(m)}(t))$. The expectation of $E(y_i|\mathbf{O}, \boldsymbol{\Theta}^{(\mathbf{m})})$ can be written as

$$w_i^{(m)} = E(y_i|\mathbf{O}, \boldsymbol{\Theta}^{(\mathbf{m})}) = \delta_i + (1 - \delta_i) \frac{\pi(\mathbf{z}_i) S(\log(t_i) - \boldsymbol{\beta}\mathbf{x_i})}{1 - \pi(\mathbf{z_i}) + \pi(\mathbf{z}_i) S(\log(t_i) - \boldsymbol{\beta}\mathbf{x_i})} \Bigg|_{(\mathbf{O}, \boldsymbol{\Theta}^{(\mathbf{m})})}. \tag{5.3}$$

Therefore, the estimation equations can be written as

$$E(l_{c_1}) = \sum_{i=1}^{n} w_i^{(m)} \log[\pi(\mathbf{z_i})] + (1 - w_i^{(m)}) \log[1 - \pi(\mathbf{z_i})], \tag{5.4}$$

$$E(l_{c_2}) = \sum_{i=1}^{n} \delta_i \log[w_i^{(m)} h(\log(t_i) - \boldsymbol{\beta}\mathbf{x_i})] + w_i^{(m)} \log[S(\log(t_i) - \boldsymbol{\beta}\mathbf{x_i})]. \tag{5.5}$$

The M-step in the EM algorithm is to maximize (2.6) and (2.7) with respect to the unknown parameters. Similar to the PH mixture cure model, the parameters in equation (5.4) can be easily estimated by '`glm`' package in R.

Zhang and Peng [31] proposed a rank-based estimation method to estimate $\boldsymbol{\beta}$ in the M-step for the semiparametric AFT mixture cure model. They turned the estimation equation (5.5) into a log-likelihood function of a standard semiparametric AFT model, except for the constant term $w_i^{(m)}$, which is

$$\log \prod_{i=1}^{n} [w_i^{(m)} h(\log(t_i) - \boldsymbol{\beta}\mathbf{x}_i)]^{\delta_i} [S(\log(t_i) - \boldsymbol{\beta}\mathbf{x}_i)^{w_i^{(m)}}].$$

This enables us to estimate $\boldsymbol{\beta}$ in the M-step by the existing semiparametric estimation method for the AFT model [19]. Zhang and Peng [31] suggested to obtain the estimator by maximizing the convex function $G(\boldsymbol{\beta})$, where

$$G(\boldsymbol{\beta}) = n^{-1} \sum_{i=1}^{n} \sum_{j=1}^{n} \delta_i w_j |\varepsilon_i - \varepsilon_j| I(\varepsilon_i - \varepsilon_j). \tag{5.6}$$

Therefore, maximization of (5.5) can be realized by maximizing (5.6) through the linear programming method in `R`.

Estimation of the Survival Function in the E-Step

Let $\tau_1 < \tau_2 < \cdots < \tau_k$ be the distinct uncensored failure residuals, which is $\log t_i - \boldsymbol{\beta}\mathbf{x}_i$, $i = 1, \cdots, n$, d_{τ_j} denote the number of failures and $R(\tau_j)$ denote the risk set at τ_j. An estimator of $S_0(\varepsilon|Y = 1)$ is given by

$$\hat{S}_0(\varepsilon|Y = 1) = \exp\left(- \sum_{j:\tau_j < \varepsilon} \frac{d_{\tau_j}}{\sum_{i \in R(\tau_j)} w_i^{(m)}} \right). \tag{5.7}$$

Same as the semiparametric PH mixture cure model, we set $\hat{S}_0(\varepsilon|Y = 1) = 0$ for $\varepsilon > \tau_k$. Then $\hat{S}(t|Y = 1) = \hat{S}_0(\varepsilon|Y = 1)$.

Variance Estimation

Because of the complexity of the estimating equation in the EM algorithm, the second derivative of estimation equation is not available. The same bootstrap method as the PH mixture cure model is used to estimate the variance of AFT mixture cure model.

5.2 Simulation Study

In the simulation study, the probability of cure is generated from a logistic model where $\pi(\mathbf{z}) = \frac{\exp(\mathbf{bz})}{1+\exp(\mathbf{bz})}$. The covariate $\mathbf{z}$ is fixed by design and is binary. Censoring times are generated from a uniform distribution $U(c_1, c_2)$, where c_1 and c_2 are some constants. The results below are based on $n = 200$ and $n = 500$ respectively with 500 replications.

Table 5.1 Estimates from Logistic-Extreme AFTMC model (2,-1,0,2)

Censoring Distribution	Censoring Rate	Parameter	True Values	n = 200 Bias	n = 200 MSE	n = 500 Bias	n = 500 MSE
U(0.5,30)	28.8	$\hat{b}_0$	2	0.0305	0.1090	0.0252	0.0454
		$\hat{b}_1$	-1	0.0657	0.3608	0.0211	0.1095
		$\hat{\beta}_0$	0	-0.6916	0.5272	-0.6904	0.5280
		$\hat{\beta}_1$	2	0.0057	0.0711	0.0142	0.0313
U(0.5,9)	42.8	$\hat{b}_0$	2	0.0720	0.3389	0.0355	0.0569
		$\hat{b_1}$	-1	-0.0801	1.1891	-0.1434	0.4802
		$\hat{\beta}_0$	0	-0.6680	0.5588	-0.6647	0.5360
		$\hat{\beta}_1$	2	-0.1093	0.2394	-0.0990	0.1054
U(0.5,5)	50.8	$\hat{b}_0$	2	0.1496	1.1778	0.0088	0.1745
		$\hat{b}_1$	-1	-0.8977	2.4754	-0.7173	0.9915
		$\hat{\beta}_0$	0	-0.5143	0.3816	-0.5584	0.3956
		$\hat{\beta}_0$	2	-0.4277	0.4993	0.0982	0.0328

In Table 5.1, $\boldsymbol{b_0} = 2$ and $\boldsymbol{b_1} = -1$ correspond to $\pi(\mathbf{z} = 0) = 0.88$ and $\pi(\mathbf{z} = 1) = 0.73$ which mean that 12% of the population is cured in control group and 27% in treatment group. Table 5.1 presents the estimated biases and MSE from the AFT mixture cure model of three regression parameters $\boldsymbol{b_0}$, $\boldsymbol{b_1}$ and $\boldsymbol{\beta}$ based on logistic-Extreme data. The error distribution follows extreme distribution. The bias are quite small with censoring rates change from light censoring (20%), moderate censoring (40%), to heavy censoring (50%). Same as the PH mixture cure model, the MSE get small when the sample size increase from 200 to 500.

5.3 Application

To illustrate the usage of `smcure` R package for semiparametric AFT mixture cure model, we fit the bone marrow transplant study for the refractory acute lymphoblastic leukemia patients as an example. The semiparametric AFT mixture cure model can be fitted as following:

```
> bmtfit <- smcure(Surv(Time,Status)~TRT,cureform=~TRT,
          data=bmt,model="aft",nboot=200)
```

The output is:

```
> printsmcure(bmtfit)
Call:
smcure(formula = Surv(Time, Status) ~ TRT, cureform = ~TRT,
   data = bmt, model = "aft", nboot = 200)


Cure probability model:
            Estimate Std.Error   Z value     Pr(>|Z|)
(Intercept) 1.007354 0.2261408 4.4545448 8.407136e-06
TRT         0.427327 0.4843662 0.8822394 3.776474e-01


Failure time distribution model:
              Estimate Std.Error   Z value  Pr(>|Z|)
(Intercept)  0.2101563 0.1783968  1.178027 0.2387859
TRT         -0.3531250 0.2705977 -1.304982 0.1918991
```

The standard errors of estimated parameters are obtained based on 200 bootstrap samples. The cure rate can be calculated based on the results from `Cure probability model` part. For example, the cure rate for the autologous transplant is 19.2 percent,

which is calculated by $1 - \hat{\pi}(\mathbf{z}) = 1 - e^{1.007354+0.427327}/(1 + e^{1.007354+0.427327})$. The estimated survival curves with respect to the treatment can be obtained by

```
> predbmt=predictsmcure(bmtfit,newX=c(0,1),newZ=c(0,1),model="aft")
> plotpredictsmcure(predbmt,model="aft")
```

From the fitted survival curves in Figure 5.1, we can see that the patients from the allogeneic treatment group has better survival probability than those from the autologous treatment group.

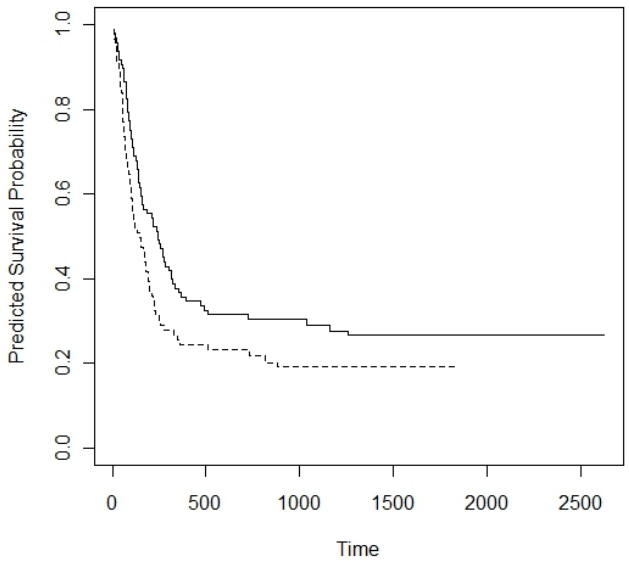

Figure 5.1 Predicted Survival curves by treatment groups for bone marrow transplant study. The upper solid line is the allogeneic treatment group and lower dashed line is the autologous treatment group.

5.4 Conclusions

We develop an R package to estimate the semiparametric PH mixture cure and AFT mixture cure models. The cure probability part is estimated by the generalized linear

model which allows many link functions, such as `logit`, `probit` and `cloglog`. The latency part can follow either the PH model or the AFT model. The semiparametric estimation procedures are based on the EM algorithm for both models. This package is an extension of the S-PLUS package `semicure` by Y. Peng which is for the PH mixture cure model only, and the SAS macro `PSPMCM` [8] which accounts for the PH mixture cure model and the parametric approach for the AFT mixture cure model. The `smcure` package in R is developed for implementing the semiparametric estimation methods to both the PH mixture cure model and the AFT mixture cure model.

Chapter 6

Summary and Conclusions

This dissertation is about advanced methodology development in mixture cure models. It consists of three projects: (1) Software Development for Estimating Semiparametric Mixture Cure Models (2) New Program of Sample Size Estimation with Cure Fraction in R (3) New Estimation Method for Semiparametric Mixture Cure Model with Competing Risks Data.

Modern medical treatments have substantially improved cure rates for many chronic diseases and have generated increasing interest in appropriate statistical models to handle survival data with non-negligible cure fractions. The mixture cure model is designed to model such data, which assumes that studied population is a mixture of individuals who are cured and individuals who are not cured. The mixture cure model assumes that a fraction of the survivors are cured from the disease of interest. The failure time distribution for the uncured individuals (latency) can be modeled by either parametric models or a semi-parametric proportional hazards model. A straightforward way to identify whether a particular dataset might have a proportion of long-term survivors is to look at the estimated survival curve. If the Kaplan-Meier survival curve has a plateau at the end of the study, a cure model may be an appropriate and useful way to analyze the data. Some statistical research has been done on mixture cure models, but none of the proposed statistical approaches has software available for public use.

In this dissertation, I first develop an R package named `smcure` to estimate the semiparametric proportional hazards (PH) mixture cure model and accelerated failure

time (AFT) mixture cure model. The cure probability part is estimated by the generalized linear model which allows many link functions, such as `logit`, `probit` and `cloglog`. The latency part can follow either the PH model or the AFT model. The semiparametric estimation procedures are based on the EM algorithm for both models. This package is an extension of the S-PLUS package `semicure` by Y. Peng which is for the PHMC model only, and the SAS macro `PSPMCM` [8] which accounts for the PHMC model and the parametric approach for the AFTMC model.

Second, I develop another R package named `NPHMC` to estimate the sample size based on the PH mixture cure model if cure fraction exits or standard PH model if there is no cure. The package provides an important and flexible tool in sample size design in survival trial with or without cure fractions.

Competing risks data are commonly seen in medical research particularly in survival analysis. I propose a new estimation approach based on the PH mixture cure model in competing risks data framework. The estimation can be obtained from the EM algorithm.

The mixture cure model generally requires a sufficiently long follow-up and large sample sizes to identify the parameters in cure fraction and latent survival distribution for uncured individuals (Farewell, 1986) [9]. Cautious interpretation of the cure fraction estimate is needed when these is no evidence of sufficient follow-up and enough samples. Therefore, it is recommended to use the mixture cure models in situations where it is clear that a cured fraction exists and follow-up beyond the time when most events have occurred.

Work in the future may include methodology developments and possible application of cure models in real world. I will continue to maintain the two contributed packages and add more features in the programs (such as different variance estimation methods, goodness of fit, etc.) and complete simulation studies for competing risks data. Besides the application in oncology, it would be interesting to apply cure

models in other fields such as Alzheimer's disease in Neurology where failure rates are low, vaccine effectiveness, prophylactic treatments for pertussis, rabies, occupational exposures in public health and procedural interventions with adjunctive treatments in cardiovascular disease.

Bibliography

[1] J. Berkson and R.P. Gage. Survival curve for cancer patients following treatment. *Journal of the American Statistical Association*, 47:501–515, 1952.

[2] J.W. Boag. Maximum likelihood estimates of the proportion of patients cured by cancer therapy. *Journal of the Royal Statistical Society. Series B (Methodological)*, 11(1):15–53, 1949.

[3] DP Byar and SB Green. The choice of treatment for cancer patients based on covariate information. *Bulletin du cancer*, 67(4):477, 1980.

[4] C. Cai, Y. Zou, Y. Peng, and J. Zhang. smcure: An r-package for estimating semiparametric mixture cure models. *Computer Methods and Programs in Biomedicine*, 108:1255–1260, 2012.

[5] Ming-Hui Chen, David P Harrington, and Joseph G Ibrahim. Bayesian cure rate models for malignant melanoma: a case-study of eastern cooperative oncology group trial e1690. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 51(2):135–150, 2002.

[6] Ming-Hui Chen, Joseph G Ibrahim, and Debajyoti Sinha. A new bayesian model for survival data with a surviving fraction. *Journal of the American Statistical Association*, 94(447):909–919, 1999.

[7] SC Cheng, Jason P Fine, and LJ Wei. Prediction of cumulative incidence function under the proportional hazards model. *Biometrics*, pages 219–228, 1998.

[8] F. Corbière and P. Joly. A sas macro for parametric and semiparametric mixture cure models. *Computer methods and programs in biomedicine*, 85(2):173–180, 2007.

[9] Vernon T Farewell. Mixture models in survival analysis: Are they worth the risk? *Canadian Journal of Statistics*, 14(3):257–262, 1986.

[10] VT Farewell. The use of mixture models for the analysis of survival data with long-term survivors. *Biometrics*, 38(4):1041–1046, 1982.

[11] Joseph G Ibrahim, Ming-Hui Chen, and Debajyoti Sinha. Bayesian semiparametric models for survival data with a cure fraction. *Biometrics*, 57(2):383–388, 2004.

[12] Z. Jin, D.Y. Lin, L.J. Wei, and Z. Ying. Rank-based inference for the accelerated failure time model. *Biometrika*, 90(2):341–353, 2003.

[13] J. D. Kalbfleisch and R. L. Prentice. The statistical analysis of failure time data. 2002.

[14] Richard Kay. Treatment effects in competing-risks analysis of prostate cancer data. *Biometrics*, pages 203–211, 1986.

[15] J.H. Kersey, D. Weisdorf, M.E. Nesbit, T.W. LeBien, W.G. Woods, P.B. McGlave, T. Kim, D.A. Vallera, A.I. Goldman, B. Bostrom, et al. Comparison of autologous and allogeneic bone marrow transplantation for treatment of high-risk refractory acute lymphoblastic leukemia. *New England Journal of Medicine*, 317(8):461–467, 1987.

[16] J.M. Kirkwood, M.H. Strawderman, M.S. Ernstoff, T.J. Smith, E.C. Borden, and R.H. Blum. Interferon alfa-2b adjuvant therapy of high-risk resected cutaneous melanoma: the eastern cooperative oncology group trial est 1684. *Journal of Clinical Oncology*, 14(1):7, 1996.

[17] A.Y.C. Kuk and C.H. Chen. A mixture model combining logistic regression with proportional hazards regression. *Biometrika*, 1992.

[18] J. F. Lawless. Statistical models and methods for lifetime data. 2003.

[19] C.S. Li and J.M.G. Taylor. A semi-parametric accelerated failure time cure model. *Statistics in medicine*, 21(21):3235–3247, 2002.

[20] Y. Peng. Fitting semiparametric cure models. *Computational statistics & data analysis*, 41(3):481–490, 2003.

[21] Y. Peng, K.B.G. Dear, JW Denham, et al. A generalized f mixture model for cure rate estimation. *Statistics in medicine*, 17(8):813–830, 1998.

[22] Yingwei Peng and Keith B. G. Dear. A nonparametric mixture model for cure rate estimation. *Biometrics*, 56(1):237–243, 2000.

[23] Y. Ritov. Estimation in a linear regression model with censored data. *Ann. Statist.*, 18:303–328, 1990.

[24] D. Schoenfeld. The asymptotic properties of nonparametric tests for comparing survival distributions. *Biometrika*, 1981.

[25] D.A. Schoenfeld. Sample-size formula for the proportional-hazards regression model. *Biometrics*, 1983.

[26] J.P. Sy and J.M.G. Taylor. Estimation in a cox proportional hazards cure model. *Biometrics*, 56(1):227–236, 2000.

[27] J.M.G. Taylor. Semi-parametric estimation in failure time mixture models. *Biometrics*, 51:899–907, 1995.

[28] A. A. Tsiatis. Estimating regression parameters using linear rank tests for censored data. *Ann. Statist.*, 18:354–372, 1990.

[29] S. Wang, J. Zhang, and W. Lu. Sample size calculation for the proportional hazards cure model. *Statistics in medicine*, 31:3959–3971, 2012.

[30] K. Yamaguchi. Accelerated failure-time regression models with a regression model of surviving fraction: An application to the analysis of" permanent employment" in japan. *Journal of the American Statistical Association*, 87:284–292, 1992.

[31] J. Zhang and Y. Peng. A new estimation method for the semiparametric accelerated failure time mixture cure model. *Statistics in medicine*, 26(16):3157–3171, 2007.

Appendix A

Source Codes for smcure package

```
library(survival)
smrank <- function(beta,Time,X,n,w,Status){
     error <- drop(log(Time)-beta%*%t(X))
     tp <- numeric()
     for(i in 1:n){
     tp[i] <- sum(as.numeric((error[i]-error)<0)*abs(error[i]-error)*w*Status[i])
        }
    sum(tp)/n
}
smsurv <- function(Time,Status,X,beta,w,model){
    death_point <- sort(unique(subset(Time, Status==1)))
if(model=='ph') coxexp <- exp((beta)%*%t(X[,-1]))
    lambda <- numeric()
    event <- numeric()
    for(i in 1: length(death_point)){
     event[i] <- sum(Status*as.numeric(Time==death_point[i]))
     if(model=='ph')
        temp <- sum(as.numeric(Time>=death_point[i])*w*drop(coxexp))
     if(model=='aft')
        temp <- sum(as.numeric(Time>=death_point[i])*w)
        temp1 <- event[i]
     lambda[i] <- temp1/temp
        }
    HHazard <- numeric()
    for(i in 1:length(Time)){
        HHazard[i] <- sum(as.numeric(Time[i]>=death_point)*lambda)
        if(Time[i]>max(death_point))HHazard[i] <- Inf
        if(Time[i]<min(death_point))HHazard[i] <- 0
        }
   survival <- exp(-HHazard)
   list(survival=survival)
}
em <- function(Time,Status,X,Z,offsetvar,b,beta,model,link,emmax,eps)
{
```

```
    w <- Status
n <- length(Status)
if(model == "ph") s <- smsurv(Time,Status,X,beta,w,model)$survival
if(model == "aft"){
if(!is.null(offsetvar)) Time <- Time/exp(offsetvar)
error <- drop(log(Time)-beta%*%t(X))
s <- smsurv(error,Status,X,beta,w,model)$survival
}
convergence<- 1000;i <-1
while (convergence > eps & i < emmax){
uncureprob <- matrix(exp((b)%*%t(Z))/(1+exp((b)%*%t(Z))),ncol=1)
if(model == "ph"){
survival<-drop(s^(exp((beta)%*%t(X[,-1]))))}
if(model == "aft"){
    error <- drop(log(Time)-beta%*%t(X))
    survival <- s}
## E step
w <- Status+(1-Status)*(uncureprob*survival)/((1-uncureprob)+uncureprob*survival)
## M step
logistfit<- eval(parse(text = paste("glm", "(", "w~Z[,-1]",",
                family = quasibinomial(link='", link, "'",")",")",sep = "")))
update_cureb <- logistfit$coef
    if(!is.null(offsetvar))
    update_cureb <- as.numeric(eval(parse(text =
    paste("glm", "(", "w~Z[,-1]+offset(offsetvar)",",
    family = quasibinomial(link='", link, "'",")",")",sep = ""))$coef)
if(model == "ph") {
update_beta <- coxph(Surv(Time, Status)~X[,-1]+
        offset(log(w)), subset=w!=0, method="breslow")$coef
if(!is.null(offsetvar)) update_beta <- coxph(Surv(Time, Status)~X[,-1]+
        offset(offsetvar+log(w)), subset=w!=0, method="breslow")$coef
update_s <-smsurv(Time,Status,X,beta,w,model)$survival}
if(model == "aft") {
update_beta <- optim(rep(0,ncol(X)), smrank,
    Time=Time,X=X,n=n,w=w,Status=Status,method="Nelder-Mead",
    control=list(reltol=0.0001,maxit=500))$par
    update_s <- smsurv(error,Status,X,beta,w,model)$survival}
convergence<-sum(c(update_cureb-b,update_beta-beta)^2)+sum((s-update_s)^2)
b <- update_cureb
        beta <- update_beta
s<-update_s
uncureprob <- matrix(exp((b)%*%t(Z))/(1+exp((b)%*%t(Z))),ncol=1)
    i <- i+1
```

```
}
em <- list(logistfit=logistfit,b=b, latencyfit= beta,Survival=s,
           Uncureprob=uncureprob,tau=convergence)
}

#####################
##  main function  ##
#####################
smcure <- function(formula,cureform,offset=NULL,data,na.action=na.omit,
          model= c("aft", "ph"),link="logit", Var=TRUE,emmax=50,eps=1e-7,nboot=100)
{
call <- match.call()
model <- match.arg(model)
cat("Program is running..be patient...")
## prepare data
data <- na.action(data)
n <- dim(data)[1]
mf <- model.frame(formula,data)
cvars <- all.vars(cureform)
Z <- as.matrix(cbind(rep(1,n),data[,cvars]))
colnames(Z) <- c("(Intercept)",cvars)
if(!is.null(offset)) {
offsetvar <- all.vars(offset)
offsetvar<-data[,offsetvar]}
else offsetvar <- NULL
Y <- model.extract(mf,"response")
X <- model.matrix(attr(mf,"terms"), mf)
if (!inherits(Y, "Surv")) stop("Response must be a survival object")
    Time <- Y[,1]
    Status <- Y[,2]
bnm <- colnames(Z)
nb <- ncol(Z)
if(model == "ph") {
betanm <- colnames(X)[-1]
nbeta <- ncol(X)-1}
if(model == "aft"){
betanm <- colnames(X)
nbeta <- ncol(X)}
## initial value
w <- Status
b <- eval(parse(text = paste("glm", "(", "w~Z[,-1]",",
          family = quasibinomial(link='", link, "'",")",")",sep = "")))$coef
if(model=="ph") beta <- coxph(Surv(Time, Status)~X[,-1]+
```

```
          offset(log(w)), subset=w!=0, method="breslow")$coef
if(model=="aft") beta <- survreg(Surv(Time,Status)~X[,-1])$coef
      ## do EM algo
emfit <- em(Time,Status,X,Z,offsetvar,b,beta,model,link,emmax,eps)
b <- emfit$b
beta <- emfit$latencyfit
s <- emfit$Survival
logistfit <- emfit$logistfit
  if(Var){
   if(model=="ph") {b_boot<-matrix(rep(0,nboot*nb), nrow=nboot)
beta_boot<-matrix(rep(0,nboot*(nbeta)), nrow=nboot)
iter <- matrix(rep(0,nboot),ncol=1)}

   if(model=="aft") {b_boot<-matrix(rep(0,nboot*nb), nrow=nboot)
beta_boot<-matrix(rep(0,nboot*(nbeta)), nrow=nboot)}
  tempdata <- cbind(Time,Status,X,Z)
  data1<-subset(tempdata,Status==1);data0<-subset(tempdata,Status==0)
  n1<-nrow(data1);n0<-nrow(data0)
  i<-1
while (i<=nboot){
             id1<-sample(1:n1,n1,replace=TRUE);id0<-sample(1:n0,n0,replace=TRUE)
bootdata<-rbind(data1[id1,],data0[id0,])
bootZ <- bootdata[,bnm]
if(model=="ph") bootX <- as.matrix(cbind(rep(1,n),bootdata[,betanm]))
if(model=="aft") bootX <- bootdata[,betanm]
bootfit <- em(bootdata[,1],bootdata[,2],bootX,bootZ,offsetvar,b,beta,model,link,emmax,eps)
b_boot[i,] <- bootfit$b
    beta_boot[i,] <- bootfit$latencyfit
    if (bootfit$tau<eps) i<-i+1}
b_var <- apply(b_boot, 2, var)
beta_var <- apply(beta_boot, 2, var)
b_sd <- sqrt(b_var)
beta_sd <- sqrt(beta_var)
}
fit<-list()
class(fit) <- c("smcure")
fit$logistfit <- logistfit
fit$b <- b
fit$beta <- beta
if(Var){
fit$b_var <- b_var
fit$b_sd <- b_sd
fit$b_zvalue <- fit$b/b_sd
```

```
fit$b_pvalue <- (1-pnorm(abs(fit$b_zvalue)))*2
fit$beta_var <- beta_var
fit$beta_sd <- beta_sd
fit$beta_zvalue <- fit$beta/beta_sd
fit$beta_pvalue <- (1-pnorm(abs(fit$beta_zvalue)))*2 }
cat(" done.\n")
fit$call <- call
fit$bnm <- bnm
fit$betanm <- betanm
fit$s <- s
fit$Time <- Time
if(model=="aft"){
error <- drop(log(Time)-beta%*%t(X))
fit$error <- error}
fit
printsmcure(fit,Var)
}
coefsmcure <- function(x, ...)
{
coef <- c(x$b,x$beta)
  names(coef) <- c(x$bnm,x$betanm)
  coef
}
printsmcure <- function(x,Var=TRUE, ...)
{
if(is.null(Var)) Var=TRUE
if(!is.null(cl <- x$call)) {
cat("Call:\n")
dput(cl)
}
cat("\nCure probability model:\n")
if (Var) {
b <- array(x$b,c(length(x$b),4))
rownames(b) <- x$bnm
      colnames(b) <- c("Estimate","Std.Error","Z value","Pr(>|Z|)")
b[,2] <- x$b_sd
b[,3] <- x$b_zvalue
b[,4] <- x$b_pvalue}

  if (!Var) {
b <- array(x$b,c(length(x$b),1))
rownames(b) <- x$bnm
      colnames(b) <- "Estimate"
```

```
}
print(b)
cat("\n")

cat("\nFailure time distribution model:\n")
if (Var) {
beta <- array(x$beta,c(length(x$beta),4))
rownames(beta) <- x$betanm
colnames(beta) <- c("Estimate","Std.Error","Z value","Pr(>|Z|)")
beta[,2] <- x$beta_sd
beta[,3] <- x$beta_zvalue
beta[,4] <- x$beta_pvalue}
if (!Var) {
beta <- array(x$beta,c(length(x$beta),1))
rownames(beta) <- x$betanm
      colnames(beta) <- "Estimate"
}
      print(beta)
invisible(x)
}
predictsmcure <- function(object, newX, newZ,model=c("ph","aft"), ...)
{
call <- match.call()
if(!inherits(object, "smcure")) stop("Object must be results of smcure")
if(is.vector(newZ)) newZ=as.matrix(newZ)
newZ=cbind(1,newZ)
if(is.vector(newX)) newX=as.matrix(newX)
s0=as.matrix(object$s,ncol=1)
n=nrow(s0)
uncureprob=exp(object$b%*%t(newZ))/(1+exp(object$b%*%t(newZ)))
scure=array(0,dim=c(n,nrow(newX)))
t=array(0,dim=c(n,nrow(newX)))
spop=array(0,dim=c(n,nrow(newX)))
  if(model=='ph')
    {ebetaX=exp(object$beta%*%t(newX))
   for( i in 1:nrow(newZ))
      {scure[,i]=s0^ebetaX[i]}
     for (i in 1:n){
        for (j in 1:nrow(newX)){
      spop[i,j]=uncureprob[j]*scure[i,j]+(1-uncureprob[j])
    }
   }
     prd=cbind(spop,Time=object$Time)
```

```
 }
  if(model=='aft')
 {
  newX=cbind(1,newX)
  ebetaX=exp(object$beta%*%t(newX))
  for( i in 1:nrow(newX))
     {t[,i]=ebetaX[i]*exp(object$error)}
 for (i in 1:n){
    for (j in 1:nrow(newX)){
     spop[i,j]=uncureprob[j]*s0[i]+(1-uncureprob[j])
     }
    }
  prd=cbind(spop=spop,Time=t)
}
 structure(list(call=call,newuncureprob=uncureprob,prediction=prd),class="predictsmcure")
}

plotpredictsmcure <- function(object, type="S", xlab="Time",
               ylab="Predicted Survival Probability",model=c("ph","aft"), ...)
{
  pred <- object$prediction
  if(model=="ph"){
    pdsort <- pred[order(pred[,"Time"]),]
    if(length(object$newuncureprob)==1) plot(pdsort[,"Time"],pdsort[,1], type="S")
    else
      matplot(pdsort[,"Time"],pdsort[,1:(ncol(pred)-1)],col=1,type="S",
         lty=1:(ncol(pred)-1),xlab=xlab,ylab=ylab)
    }
  if(model=="aft"){
    nplot=ncol(pred)/2
    pdsort <- pred[order(pred[,1+nplot]),c(1,1+nplot)]
    plot(pdsort[,2],pdsort[,1],xlab=xlab,ylab=ylab,col=1,type="S",ylim=c(0,1))
    if(nplot>1){
      for(i in 2:nplot){
        pdsort<- pred[order(pred[,i+nplot]),c(i,i+nplot)]
        lines(pdsort[,2],pdsort[,1],lty=i,type="S")
        }
      }
    }
  }
```

Appendix B

Source Codes for NPHMC package

```
library(survival)
Sc<-function(t,accrualtime,followuptime,accrualdist){
 if(accrualdist=="uniform") return((accrualtime+followuptime-t)/accrualtime)
 if(accrualdist=="increasing") return((accrualtime+followuptime-t)^2/accrualtime^2)
 if(accrualdist=="decreasing") return((1-(followuptime-t)^2/accrualtime^2))
}

f1<-function(t,survdist,k,lambda0){
 if(survdist=="exp") {k=1; return(lambda0*k*(lambda0*t)^(k-1)*exp(-(lambda0*t)^k))}
 if(survdist=="weib") {return(lambda0*k*(lambda0*t)^(k-1)*exp(-(lambda0*t)^k))}
}

f2<-function(t,accrualtime,followuptime,accrualdist,survdist,k,lambda0){
 Sc(t,accrualtime,followuptime,accrualdist)*f1(t,survdist,k,lambda0)
}

H0<-function(t,survdist,k,lambda0){
 if(survdist=="exp") {return(lambda0*t)}
 if(survdist=="weib") {return((lambda0*t)^k)}
}

S0<-function(t,pi0,survdist,k,lambda0){
 if(survdist=="exp") {k=1;return(pi0+(1-pi0)*exp(-(lambda0*t)^k))}
 if(survdist=="weib") {return(pi0+(1-pi0)*exp(-(lambda0*t)^k))}
}

m<-function(t,beta0,gamma0,pi0,survdist,k,lambda0){
 (gamma0/beta0+H0(t,survdist,k,lambda0))*pi0/S0(t,pi0,survdist,k,lambda0)-1}

f3<-function(t,beta0,gamma0,pi0,survdist,k,lambda0){
 m(t,beta0,gamma0,pi0,survdist,k,lambda0)*f1(t,survdist,k,lambda0)
}

f4<-function(t,accrualtime,followuptime,accrualdist,beta0,gamma0,pi0,survdist,k,lambda0){
```

```
 m(t,beta0,gamma0,pi0,survdist,k,lambda0)*
 f2(t,accrualtime,followuptime,accrualdist,survdist,k,lambda0)
}

NPHMC<-function(power=0.8,alpha=0.05,accrualtime,followuptime,p=0.5,
        accrualdist=c("uniform","increasing","decreasing"),
hazardratio,oddsratio,pi0,survdist=c("exp","weib"),k=1,lambda0,data=NULL){
n<-list()
class(n) <- c("NPHMC")
if (is.null(data)){
      if (hazardratio<=0) stop("Hazardratio must be greater than 0")
if (oddsratio<0) stop("Oddsratio cannot be less than 0")
        if (pi0==0 | oddsratio==0) {
            i1 <- integrate(f1,0,followuptime,survdist,k,lambda0)$value
 i2 <- integrate(f2,followuptime,(accrualtime+followuptime),accrualtime,followuptime,
          accrualdist,survdist,k,lambda0)$value
      beta0 <- log(hazardratio)
          pdeath <- i1+i2
      nsizeph <- ceiling((qnorm(power)-qnorm(alpha/2))^2/(p*(1-p)*beta0^2*pdeath))
      cat("========================================================================== \n")
  cat("SAMPLE SIZE CALCULATION BASED ON STANDARD PH MODEL (NO CURE FRACTION) \n")
  cat("========================================================================== \n")
      cat("Standard PH Model: n =",nsizeph,"\n")}
        else {
  i1 <- integrate(f1,0,followuptime,survdist,k,lambda0)$value
  i2 <- integrate(f2,followuptime,(accrualtime+followuptime),
           accrualtime,followuptime,accrualdist,survdist,k,lambda0)$value
  beta0 <- log(hazardratio)
  gamma0 <- log(oddsratio)
  i3 <- integrate(f3,0,followuptime,beta0,gamma0,pi0,survdist,k,lambda0)$value
  i4 <- integrate(f4,followuptime,(accrualtime+followuptime),
      accrualtime,followuptime,accrualdist,beta0,gamma0,pi0,survdist,k,lambda0)$value
nsize <- ceiling((qnorm(power)-
            qnorm(alpha/2))^2*(i1+i2)/((i3+i4)^2*p*(1-p)*(1-pi0)*beta0^2))
pdeath <- i1+i2
nsizeph <- ceiling((qnorm(power)-qnorm(alpha/2))^2/(p*(1-p)*beta0^2*pdeath))
cat("\n")
cat("========================================================================== \n")
cat("SAMPLE SIZE CALCULATION FOR PH MIXTURE CURE MODEL AND STANDARD PH MODEL \n")
cat("========================================================================== \n")
cat("PH Mixture Cure Model: n =",nsize,"\n")
cat("Standard PH Model: n =",nsizeph,"\n")
      n$nsize <- nsize }
```

```
     }
  if (!is.null(data)){

ta=accrualtime
tf=followuptime
ttot=ta+tf
t<-data[,1]
      colnames(data)<-c("Time","Status","X")
Time=data[,1]
Status=data[,2]
X=data[,3]
      f=smcure(Surv(Time, Status)~X,~X,data=data,model="ph",Var=FALSE)
time<-sort(t[Status==1])
beta0nocure <- coxph(Surv(Time, Status)~X,method="breslow", data=data)$coef
death_point <- sort(unique(subset(Time, Status==1)))
coxexp <- exp(beta0nocure*X)
lambda <- numeric()
     event <- numeric()
      for(i in 1: length(death_point)){
       event[i] <- sum(Status*as.numeric(Time==death_point[i]))
        temp <- sum(as.numeric(Time>=death_point[i])*Status*drop(coxexp))
        temp1 <- event[i]
       lambda[i] <- temp1/temp
        }
    HHazard <- numeric()
    for(i in 1:length(Time)){
        HHazard[i] <- sum(as.numeric(Time[i]>=death_point)*lambda)
        if(Time[i]>max(death_point))HHazard[i] <- Inf
        if(Time[i]<min(death_point))HHazard[i] <- 0
        }
    snocure <- exp(-HHazard)

beta0 <- f$beta
print(beta0)
gamma0 <- -f$b[2]
pi0=1-exp(f$b[1])/(1 + exp(f$b[1]))
    s=sort(f$s[Status==1],decreasing = TRUE)
snocure <- sort(snocure[Status==1],decreasing = TRUE)
     f0<-diff(s)

      f0nocure <- diff(snocure)

    s1 <- sum(-f0*as.numeric(time<=tf)[-1])
```

```
        s1nocure <- sum(-f0nocure*as.numeric(time<=tf)[-1])
sc=(ta+tf-time)/ta
s2 <- sum(-diff(s)*sc[-length(sc)]*as.numeric(time>tf)[-1])
        s2nocure <- sum(-diff(snocure)*sc[-length(sc)]*as.numeric(time>tf)[-1])
Spop=pi0+(1-pi0)*s
m=(gamma0/beta0-log(s))*pi0/Spop-1
    s3 <- sum(-diff(s)*m[-length(m)]*as.numeric(time<=tf)[-1])
Spop4=pi0+(1-pi0)*s
m4=(gamma0/beta0-log(s))*pi0/Spop4-1
s4 <- sum(-diff(s)*m4[-length(m4)]*sc[-length(sc)]*
          as.numeric((time>tf) & (time<=ttot))[-1])
nonpar=ceiling((qnorm(power)-qnorm(alpha/2))^2*(s1+s2)/((s3+s4)^2*p*(1-p)*(1-pi0)*beta0^2))
n$nonpar<- nonpar
n$HR <- exp(beta0)
n$OR <- exp(gamma0)
n$pi0<- pi0
cat("\n")
cat("============================================================================ \n")
cat("SAMPLE SIZE CALCULATION FOR PH MIXTURE CURE MODEL AND STANDARD PH MODEL \n")
cat("============================================================================ \n")
cat("PH Mixture Cure Model with KM estimators: n =",nonpar,"\n")
pdeathNonpar <- s1+s2
 pdeathNonpar <- s1nocure+s2nocure
#cat("Probability of Death: p =",pdeathNonpar,"\n")
nonparPH <-
    ceiling((qnorm(power)-qnorm(alpha/2))^2/(p*(1-p)*beta0nocure^2*pdeathNonpar))
cat("Standard PH Model with KM estimators: n =",nonparPH,"\n")
n$nonparPH<- nonparPH
  }
 }
```