Summer 2022

Modified EM Algorithm in SMCURE Package Based on Proportional Hazards Mixture Cure Model With Offset Terms

Jiaying Yi

Modified EM Algorithm in SMCURE Package Based on proportional hazards mixture cure Model with Offset Terms

by

Jiaying Yi

Bachelor of Science
China Agricultural University 2019

Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in

Biostatistics

The Norman J. Arnold School of Public Health

University of South Carolina

2022

Accepted by:

Jiajia Zhang, Director of Thesis

Robert Moran, Reader

Yuan Wang, Reader

Tracey L. Weldon, Vice Provost and Dean of the Graduate School

Acknowledgments

First, I would like to express my gratitude to my thesis advisor, Dr. Jiajia Zhang, who has provided me with the topic of the thesis and supported me to continue the thesis research and writing. The course survival analysis by her has prepared me strong theoretical basis and her further research instructions and discussion with great patience gave me the confidence and inspiration for the thesis.

My thesis committee members, Dr. Yuan Wang and Dr. Robert Moran have also provided valuable suggestions to my thesis, including evaluating the numerical study results and helping me improve the thesis.

Last but not certainly least, I must thank my parents and my friend. Without their help I could not make it through the pandemic outbreak and finished my study abroad.

Abstract

Mixture cure model is a useful method of survival analysis for population including cured proportion and uncured proportion. The R package SMCURE applies EM algorithm to estimate the coefficients of covariates in the mixture cure model. Although an offset term is specified in the SMCURE statement, the offset term is not appropriately handled in the algorithm. This thesis aims to adjust the EM algorithm for the proportional hazards mixture cure model in the SMCURE package. In addition, the offset term can be specified separately in the incidence part or the latency part. The numerical experiments include simulation study and real data application on the bone marrow transplantation data, and the results indicate that the modified EM algorithm for the proportional hazards mixture cure model with offset term generates smaller bias and variance estimation, compared to the proportional hazards mixture cure model without considering offset terms.

Table of Contents

Acknowledgments iii

Abstract iv

List of Tables vii

List of Figures viii

Chapter 1 Introduction 1

1.1 Background 1

1.2 Mixture cure model 2

1.3 Research problem and aim 4

1.4 Thesis outline 4

Chapter 2 Proportional hazards mixture cure model 5

2.1 Proportional hazards mixture cure model 5

2.2 EM algorithm for the proportional hazards mixture cure model in the SMCURE package 6

2.3 Modified EM algorithm for the proportional hazards mixture cure model 10

2.4 Variance estimation 14

Chapter 3 Numerical Experiment 16

3.1 Simulation study 16

3.2 Real data study 26

Chapter 4 Summary and Future work 31

Bibliography 33

Appendix A R code for the modified EM algorithm 36

Appendix B R code for first simulation study design with Weibull 1 data set 61

Appendix C R code for BMT data analysis 65

List of Tables

Table 3.1 Simulation data settings for different censoring rates 18

Table 3.2 Simulation results for Weibull data set 19

Table 3.3 Simulation results for log-Normal data set 20

Table 3.4 Simulation results for exponential data set 20

Table 3.5 Average iteration times using modified EM algorithm 21

Table 3.6 The cure rates and censoring rates of four types of parameter settings 22

Table 3.7 Simulation results of bias and variance with x_3 both as the incidence offset and the latency offset term 23

Table 3.8 Simulation results of bias and variance with x_2 both as the incidence offset and the latency offset term 24

Table 3.9 Simulation results of bias and variance with x_2 as the incidence offset and x_3 as the latency offset 25

Table 3.10 Simulation results of bias and variance with x_3 as the incidence offset and x_2 as the latency offset 25

Table 3.11 Average iteration times using modified EM algorithm 26

Table 3.12 Covariates information in BMT data 27

Table 3.13 Results of SMCURE for all risk factors 29

Table 3.14 Results from two algorithms for BMT data 30

List of Figures

Figure 3.1 Kaplan-meier survival curve for BMT data 27

Figure 3.2 Kaplan-meier survival curves for BMT data in groups with and without MTX 28

Figure 3.3 Estimated survival curves for patients using MTX or not 30

Chapter 1

Introduction

1.1 Background

Survival analysis is stated as statistical procedures for data analysis for time to event data in some specific population (Kleinbaum et al. (2012)). Thus, it is also called "time to event" analysis. Due to the restriction of the study period, resources and other potential reasons of loss to follow up, there exists a variety of censoring scheme in survival data. In the literature, many studies focus on estimating risk factors to survival probability, under the different censoring settings such as the right censoring, interval censoring, left censoring and current status data.

The most common standard survival models, inclusing the proportional hazards (PH) model, the accelerated failure time (AFT) model and the proportional odds model, assume that patients will eventually experience the event of interest as long as the follow up time is long enough. This assumption indicates that all patients will die from the event of interest as long as the follow up time is adequate. However, this is not always true in practice. With the advancements in medical research, many diseases are curable such as early stage of breast cancer, prostate cancer and colon cancer. In the case of curable disease, some patients could be considered as "cured", which means those patients will not experience the event of interest even though the follow up time is adequate. The mixture cure model, proposed by Boag (1949) and Berkson and Gage (1952), was designed to handle the survival data with a curable fraction. There are two components in the mixture cure model. One is referred to as a cured component and the other one is referred to as the uncured component.

For the cured patients, the model will provide the cure rate estimation; while for the uncured component, the model will further examine the survival probability of uncured patients.

1.2 Mixture cure model

The two components in the mixture cure model are modelled via an uncured probability and a standard survival distribution (Berkson and Gage (1952)). Specifically, the model is

$$S_{pop}(t) = \pi S(t) + (1 - \pi) \tag{1.1}$$

where $S_{pop}(t)$ is the survival function of the people in population, π is the probability of being uncured, and $S(t)$ is the survival function for uncured proportion. It is common to refer π as the incidence part and $S(t)$ as the latency part.

The estimation method for Equation (2.1) started with a fully parametric methods, semi-parametric methods, and non-parametric methods. The covariates were been added to the mixture cure model to reveal the risk factors to the incidence and the latency parts. The mixture cure model proposed by Boag (1949) and Berkson and Gage (1952) was fully parametric without covariates. Farewell and Prentice (1977) first introduced covariates into the incidence part via a logistic regression, and kept the latency part as an exponential distribution without covariates. Farewell (1982) added covariates to the latency via a Weibull distribution and still kept incidence as a logistic regresssion. Yamaguchi (1992) discussed two types of incidence, including a constant or a logistic regression, and applied the AFT model for the latency with an extended Gamma family. Ghitany et al. (1994) improved the mixture cure model by considering the covariates in both incidence a d latency parts via the logistic regression and the exponential distribution. Peng et al. (1998) considered the similar mixture cure model with the AFT in the latency with a generalized F distribution, which includes the exponential, Weibull, log-Normal distributions and so on.

The fully parametric estimation approach are not flexible enough for some situations since the parametric distribution is hard to specify in practice. Thus, there are many discussions on more flexible semi-parametric estimation methods. The most popular mixture cure model is the proportional hazards mixture cure model which applies a Proportional Hazards (Cox PH) model (Cox (1972)) to the latency part while remaining the logistic link function in the incidence part. Kuk and Chen (1992) used the Monte Carlo approximation to maximize the marginal likelihood (Kalbfleisch and Prentice (1973)) for the regression parameters, and applied EM algorithm (Dempster et al. (1977)) for the baseline survival functions in the latency part. Sy and Taylor (2000) used the Breslow estimator (Breslow (1974)) to estimate the baseline conditional cumulative hazard function, and the latency part could therefore be estimated via the partial likelihood estimation method as in the classical PH model. The estimation algorithms of the PH mixture cure model in peng2000nonparametric is similar to the one from Kuk and Chen (1992), and the methods include EM algorithm, marginal likelihood and multiple imputations for parameter estimation. Lu (2008) proposed a non-parametric maximum likelihood estimator to estimate the cumulative hazard function and regression parameters. Lam et al. (2005) considered simple multiple imputation method for the cured proportion. Wang et al. (2012) used smoothing splines analysis of variance for both the incidence and the latency part. Note, when the PH assumptions were violated for the uncured patients, the PH model could not be applied for the latency part.

The alternative mixture cure model includes the AFT mixture cure model (Li and Taylor, 2002), which was estimated via the EM algorithm. López-Cheda et al. (2017) proposed a completely non-parametric model, which was based on the Beran estimator (Beran (1981)) for estimators both in the incidence part and the latency part. Additionally, a bootstrap bandwidth selection method was applied for the non-parametric estimator for the incidence part.

1.3 Research problem and aim

The SMCURE package in R by Cai et al. (2012) was developed to estimate coefficients in the mixture cure model including the PH mixture cure model and the AFT mixture cure model using the semi-parametric approach. Specifically, it assumed a logistic link for the incidence and the PH model or AFT model for the latency part. The EM algorithm was adopted in the estimation due to the unknown latent cure indicators, and the variance estimation was accomplished via the bootstrap method. However, even there was an option of offset term in the package for the incidence part and the latency part, the SMCURE function did not adjust the offset terms correctly in all EM steps and could not produce a correct estimation under the offset option. Therefore, this thesis aims to modify the computation algorithm for the PH mixture cure model with offset terms separately in the incidence and the latency part, and to evaluate the performance of the modified algorithm based on comprehensive simulation studies and real data analysis.

1.4 Thesis outline

Chapter 2 states the PH mixture cure model, the algorithms used in SMCURE package, the modified EM algorithm for the PH mixture cure model with the offset terms separately in the incidence and the latency part, and the bootstrap method for the variance estimation. Chapter 3 evaluates the performance of the modified algorithm using comprehensive simulation studies and real data analysis. The results are reported in tables and figures. Chapter 4 summarizes the thesis and future research direction. Codes for the modified EM algorithm and additional simulation studies are presented in the appendices.

Chapter 2

Proportional hazards mixture cure model

In this chapter, we provide an overview of the proportional hazards mixture cure (PHMC) model in Section 2.1. The EM algorithm from the SMCURE package for the PHMC model is then stated in Section 2.2; the modified EM algorithm with offset terms in either incidence, latency or both components is in Section 2.3. Finally, the bootstrap method is presented for variance estimation in Section 2.4.

2.1 Proportional hazards mixture cure model

The mixture cure model with coviariate is expressed as

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

where $S_{pop}(t|z,x)$ is a population survival, $\pi(z)$ is the incidence part with a covariate vector z and $S_(t|x)$ is the survival probability for the uncured patients with a covariate vector x.

The PHMC model assumes the logistic regression for the incidence part and the proportional hazards model for the latency part. Specifically,

$$\pi(z) = \frac{\exp(b^T z)}{1 + \exp(b^T z)} \tag{2.2}$$

where b is an unknown parameter vector for the covariate vector z. The latency part under the proportional hazards model is

$$S(t|x) = S_0(t)^{\exp(x^T \beta)} \tag{2.3}$$

where $S_0(t)$ is a baseline survival function, and β is the unknown parameter vector for the covariate vector x. The unknown parameters in the PHMC model is denoted as $\Theta = (b, \beta, S_0(t))$.

Suppose there are N patients, and for the i^{th} patient the observations can be written as $O_i = (T_i, \delta_i, z_i, x_i) \in O$ $(i = 1, 2, \cdots, N)$, where T_i is the observed survival time (the minimum of the event time and the censoring time), δ_i is a censoring indicator with $\delta_i = 0$ for the censored time and 1 for the event time, and z_i, x_i are covariate vectors respectively for the incidence and latency part. Then, we have the observed likelihood function

$$L_{obs}(b, \beta, S_0(t); O) = \prod_{i=1}^{N} \left[\pi(z_i) f(t|x_i)\right]^{\delta_i} \times \prod_{i=1}^{N} \left[(1 - \pi(z_i)) + \pi(z_i) S(t|x_i)\right]^{1-\delta_i} \quad (2.4)$$

where $f(t|x_i)$ is the probability density function for being uncured, corresponding to the survival function of uncured patients, $S(t|x_i)$.

2.2 EM algorithm for the proportional hazards mixture cure model in the SMCURE package

Let us introduce a latent variable y_i, which is a cure indicator for patient i with $y_i = 1$ for uncured and 0 for cured. If a patient has the event of interest ($\delta_i = 1$), the patient must experience the event of interest, which indicates he/she is uncured with $y_i = 1$. If the data for the patient is censored ($\delta_i = 0$), the cure status of this person is unknown ($y_i = 0$ or 1). Given such partially known cure indicator y_i, the complete likelihood function therefore can be

$$\begin{aligned} L(b, \beta, S_0(t); O) = & \prod_{i=1}^{N} \left[\pi(z_i) h(t_i|x_i) S(t_i|x_i)\right]^{\delta_i y_i} \times \\ & \prod_{i=1}^{N} \left[\pi(z_i) S(t_i|x_i)\right]^{(1-\delta_i) y_i} \times \prod_{i=1}^{N} \left[1 - \pi(z_i)\right]^{1-y_i} \end{aligned} \quad (2.5)$$

$h(t_i|x_i)$ is the hazard function corresponding to $S(t|x_i)$ with relationship $f(t_i|x_i) = h(t_i|x_i)S(t_i|x_i)$. Equation (2.5) can be simplified as

$$L(b, \beta, S_0(t); O) = \prod_{i=1}^{N} [1 - \pi(z_i)]^{1-y_i} \pi(z_i)^{y_i} h(t_i|x_i)^{\delta_i y_i} S(t_i|x_i)^{y_i} \tag{2.6}$$

Note, $L(b, \beta, S_0(t); O)$ can be written as a product of two functions with the first component with respect to unknown parameter b and the second component with respect to $\beta, S_0(t)$.

We further calculate the log-likelihood function denoted as $l(b, \beta; O_i)$,and we expressed as follows.

$$l(b, \beta, S_0(t); O) = l_1(b; O) + l_2(\beta, S_0(t); O) \tag{2.7}$$

where

$$l_1(b; O) = \sum_{i=1}^{N} \{y_i \log[\pi(z_i)] + (1 - y_i) \log[1 - \pi(z_i)]\} \tag{2.8}$$

$$l_2(\beta, S_0(t); O) = \sum_{i=1}^{N} \{y_i \delta_i \log[h(t_i|x_i)] + y_i \log[S(t_i|x_i)]\} \tag{2.9}$$

Since the cure indicator y_i is a latent variable, the EM algorithm should be considered for the estimation of b, β and $S_0(t)$.

The EM algorithm mainly contains two steps: E-step and M-step. The E-step calculates the expectation of the complete likelihood with respect to y_i under current estimated $\hat{\Theta}^{(m)}$. Because the probability of being uncured is estimated under the logistic regression,the conditional distribution of y_i follows a Binomial distribution. Let w_i be the conditional expectation of y_i, and it can be written as

$$\begin{aligned} w_i^{(m)} &= E(y_i|O, \hat{\Theta}^{(m)}) \\ &= \delta_i + (1 - \delta_i) \frac{\pi(z_i) S(t_i|x_i)}{1 - \pi(z_i) + \pi(z_i) S(t_i|x_i)} \Big|_{O, \hat{\Theta}^{(m)}} \end{aligned} \tag{2.10}$$

Note, $w_i^{(m)}$ indicates the probability of being uncured. When $\delta_i = 1$, $w_i^{(m)} = 1$, and a probability of $w_i^{(m)} \in (0, 1)$ when $\delta_i = 0$.

Substituting y_i with $w_i^{(m)}$ in Equation (2.8) and Equation (2.9), we obtain

$$E(l_1(b;O)) = \sum_{i=1}^{N}\{w_i^{(m)}\log[\pi(z_i)] + (1-w_i^{(m)})\log[1-\pi(z_i)]\} \tag{2.11}$$

$$E(l_2(\beta, S_0(t);O)) = \sum_{i=1}^{N}\left[\delta_i w_i^{(m)}\log(h(t_i|x_i)) + w_i^{(m)}\log(S(t_i|x_i))\right] \tag{2.12}$$

This completes the E-step.

The M-step is to maximize the expectation of the log-likelihood function in Equation (2.11) to estimate unknown parameters. Because $\delta_i w_i = \delta_i$ and $\delta_i \log(w_i) = 0$, $E(l_2(\beta, S_0(t);O))$ can be rewritten as

$$\begin{aligned} E(l_2(\beta, S_0(t);O)) &= \sum_{i=1}^{N}\left[\delta_i\log(h(t_i|x_i)) + 0 + w_i^{(m)}\log(S(t_i|x_i))\right] \\ &= \sum_{i=1}^{N}\left[\delta_i\log(h(t_i|x_i)) + \delta_i w_i^{(m)} + w_i^{(m)}\log(S(t_i|x_i))\right] \\ &= \sum_{i=1}^{N}\left[\delta_i\log(w_i^{(m)}h(t_i|x_i)) + w_i^{(m)}\log S(t_i|x_i)\right] \end{aligned} \tag{2.13}$$

The form can be expressed in the form of log likelihood function of the PH model with an offset term $\log(w_i)$ according to the Equation 2.14.

$$\begin{aligned} E(l_2(\beta, S_0(t);O)) = \sum_{i=1}^{N}\log\Big[&\Big(h_0(t_i)\exp(x_i^T\hat{\beta}^{(m)} + \log(w_i^{(m)}))\Big)^{\delta_i} \\ &S_0(t_i)^{\exp(x_i^T\hat{\beta}^{(m)}+\log(w_i^{(m)}))}\Big] \end{aligned} \tag{2.14}$$

Therefore, the M-step can be completed via maximizing Equation (2.11) through the GLM function and maximizing Equation (2.14) by the COXPH function with an offset $\log(w_i)$.

The survival function needs to be estimated in order to conduct the E-step in the EM algorithm. Let the distinct uncensored event time be $t_{(1)} < t_{(2)} < t_{(3)} < \cdots < t_{(k)}$, the number of events be $d_{t_{(j)}}$, and the risk set at time $t_{(j)}$ be $R(t_{(j)})$. The baseline survival function is estimated by the Breslow-type estimator

$$\hat{S}_0^{(m)}(t_i) = \exp\left(-\sum_{j:t_{(j)}\le t}\frac{d_{t_{(j)}}}{\sum_{i\in R(t_{(j)})} w_i^{(m)}\exp(x_i^T\hat{\beta}^{(m)})}\right) \tag{2.15}$$

Note that the baseline survival $\hat{S}_0(t)$ do not approach to 0 as $t \to \infty$. In order to make the baseline survial as a proper distribution, we set $\hat{S}_0(t) = 0$ when $t > t_{(k)}$. Therefore, the survival function can be

$$\hat{S}^{(m)}(t_i|x_i) = \hat{S}_0(t_i)^{\exp(x_i^T \hat{\beta}^{(m)})} \tag{2.16}$$

After $\hat{b}^{(m)}$, $\hat{\beta}^{(m)}$ and $\hat{S}^{(m)}(t_i|x_i)$ are computed, they can be plugged into the E-step to update the $\hat{b}^{(m+1)}$, $\hat{\beta}^{(m+1)}$, and $\hat{S}^{(m+1)}(t_i|x_i)$.

The EM algorithm in the SMCURE package is listed in Algorithm 1. The input values include the observations matrix $O_i = (T_i, \delta_i, z_i, x_i), i = 1, \cdots, N$, the maximum number of EM iteration times n_{max}, and a convergence bound ϵ. The output values should be the estimation of $\hat{\Theta} = (\hat{b}, \hat{\beta}, \hat{S}_0(t))$ and the population survival function, which can be used to calculate $\hat{S}_{pop}(t|z_i, x_i)$.

The initial values are set as follows. First, let $w_i^{(0)} = \delta_i$. Second, we calculate $\hat{b}^{(0)}$ and $\hat{\beta}^{(0)}$ based on $w_i^{(0)}$ through the GLM function and the COXPH function in R. Third, we calculate the initial values for $\hat{S}_0^{(0)}(t_i)$ and $\hat{S}^{(0)}(t_i|x_i)$. With those initial values, we first calculate the conditional expectation of y_i, $w_i^{(m)}$, based on Equation (2.10). Then, the M-step maximizes $E(l_1(b; O))$ and $E(l_2(\beta, S_0(t); O))$ through the GLM function and the COXPH function in R. Last, the Breslow estimator for the baseline survival function is updated. The criteria to stop the iteration is 1) it meets the convergence, $|con| < \epsilon$ or 2) it reaches the maximum number of iteration times n_{max}. Here,

$$con = \sum_{i=1}^{n}[(\hat{b}^{(m)} - \hat{b}^{(m-1)})^2 + (\hat{\beta}^{(m)} - \hat{\beta}^{(m-1)})^2] + \sum_{i=1}^{n}[\hat{S}_0(t)^{(m)} - \hat{S}_0(t)^{(m-1)}]^2 \tag{2.17}$$

Algorithm 1 EM algorithm in the SMCURE Package

input $O_i = (T_i, \delta_i, z_i, x_i)$, n_{max}, ϵ, $(i = 1, 2, \cdots, N)$

output $\hat{\Theta} = (\hat{b}, \hat{\beta}, \hat{S}_0(t))$

1: Initialisation of $w_i^{(0)}$, $\hat{b}^{(0)}$, $\hat{\beta}^{(0)}$, and $\hat{S}_0(t)^{(0)}$

2: m = 1

3: E-step: calculate $\pi(z_i)$ and $w_i^{(m)}$

4: M-step:

(I) Maximize $E(l_1(b;O))$ using GLM function with quasi-binomial model to estimate $\hat{b}^{(m)}$

(II) Maximize $E(l_2(\beta, S_0(t);O))$ using COXPH function with offset $\log(w_i)^{(m)}$ to estimate $\hat{\beta}^{(m)}$

(III) Estimate baseline survival function $\hat{S}_0^{(m)}(t)$

5: Check the iteration criteria:

If $|con| < \epsilon$ and iteration times $< n_{max}$, then $m = m + 1$ and update the parameters

else stop the iteration and report output

The original algorithm in SMCURE cannot address the offset term properly, since it only considered the offset term in the $E(l_2(\beta, S_0(t);O))$ maximum step (in COXPH function statement). In addition, the offset term cannot be separately specified in incidence and latency. Therefore, a modified EM algorithm is considered.

2.3 Modified EM algorithm for the proportional hazards mixture cure model

Let u be the offset covariate in the incidence part, and v be the offset covariate in the latency part. The mixture cure model is expressed as

$$S_{pop}(t|x,z,u,v) = \pi(z,u)S(t|x,v) + (1 - \pi(z,u)) \tag{2.18}$$

Similarly, $S_{pop}(t|x,z,u,v)$ is the survival function for a patient in population with covariates x, z and offset terms u and v. $\pi(z,u)$ is the probability of being uncured for a patient depending on z and u, and $S(t|x,v)$ is the survival function of patients being uncured depending on x and v.

The probability of a patient being uncured is estimated by the logistic link function is

$$\pi(z,u) = \frac{\exp(b^T z + u)}{1 + \exp(b^T z + u)} \tag{2.19}$$

Also, the latency under the PH model is

$$S(t|x,v) = S_0(t)^{\exp(x^T\beta + v)} \tag{2.20}$$

Since the offset terms u and v have the coefficients of 1 and do not need estimation, the unknown parameter vector can still be written as $\Theta = (b, \beta, S_0(t))$ with a different length.

For a sample with a size N, the observations are $O_i = (T_i, \delta_i, z_i, x_i, u_i, v_i) \in O$ $(i = 1, 2, \cdots, N)$. For a patient i, T_i is the survival time, δ_i is a censoring indicator when $\delta_i = 0$ means censored data, z_i, u_i are covariates for incidence, and x_i, v_i are covariates for latency. The observed likelihood function is

$$\begin{aligned} L_{obs}(b,\beta,S_0(t);O) = &\prod_{i=1}^{N} [\pi(z_i,u_i) f(t_i|x_i,v_i)]^{\delta_i} \times \\ &\prod_{i=1}^{N} [(1-\pi(z_i,u_i)) + \pi(z_i,u_i) S(t_i|x_i,v_i)]^{1-\delta_i} \end{aligned} \tag{2.21}$$

$\pi(z_i, u_i)$ is the incidence and $S(t_i|x_i, v_i)$ is the latency. $f(t_i|x_i, v_i) = h(t_i|x_i, v_i)S(t_i|x_i, v_i)$ is the probability density function for uncured patients where $h(t_i|x_i, v_i)$ is a hazard function.

We also introduce a cure indicator y_i, which is a latent variable when $y_i = 1$ is an uncured event of interest. The complete likelihood function can be written as

$$L(b,\beta,S_0(t);O) = \prod_{i=1}^{N} [1-\pi(z_i,u_i)]^{1-y_i} \pi(z_i,u_i)^{y_i} h(t_i|x_i,v_i)^{\delta_i y_i} S(t_i|x_i,v_i)^{y_i} \tag{2.22}$$

The log-likelihood function can also be expressed as

$$l(b,\beta,S_0(t);O) = l_1(b;O) + l_2(\beta,S_0(t);O) \tag{2.23}$$

$$l_1(b;O) = \sum_{i=1}^{N} \{y_i \log[\pi(z_i,u_i)] + (1-y_i)\log[1-\pi(z_i,u_i)]\} \tag{2.24}$$

$$l_2(\beta, S_0(t); O) = \sum_{i=1}^{N} \{y_i \delta_i \log[h(t_i|x_i, v_i)] + y_i \log[S(t_i|x_i, v_i)]\} \tag{2.25}$$

EM algorithm is applied to deal with the latent variable y_i. The E-step is to calculate the conditional expectation of the complete log-likelihood respect to y_i, Based on a similar argument, y_i follows the Binomial distribution and the conditional expectation w_i at the m^{th} iteration step is

$$\begin{aligned} w_i^{(m)} &= E(y_i|O, \hat{\Theta}^{(m)}) \\ &= \delta_i + (1 - \delta_i) \frac{\pi(z_i, u_i) S(t_i|x_i, v_i)}{1 - \pi(z_i, u_i) + \pi(z_i, u_i) S(t_i|x_i, v_i)} \Big|_{O, \hat{\Theta}^{(m)}} \end{aligned} \tag{2.26}$$

The M-step is to estimate $\hat{b}$ and $\hat{\beta}$ via maximizing $E(l_1(b; O))$ and $E(l_2(\beta, S_0(t); O))$. Note that Equation (2.24) and Equation (2.25) both have the linear form of y_i and thus we could replace them by w_i.

$$E(l_1(b; O)) = \sum_{i=1}^{N} \Big[w_i^{(m)} \log(\pi(z_i, u_i) + (1 - w_i^{(m)}) \log(1 - \pi(z_i, u_i)) \Big] \tag{2.27}$$

$E(l_1(b; O))$ can be estimated through GLM function with a quasi-binomial option. Based on the PH model, $E(l_2(\beta; O))$ can be

$$\begin{aligned} E(l_2(\beta, S_0(t); O)) &= \sum_{i=1}^{N} \Big[\delta_i \log[w_i^{(m)} h(t_i|x_i, v_i)] + w_i^{(m)} \log[S(t_i|x_i, v_i)] \Big] \\ &= \sum_{i=1}^{N} \log \Big[\Big(h_0(t_i) \exp(x_i^T \hat{\beta}^{(m)} + v_i + \log(w_i^{(m)})) \Big)^{\delta_i} \times \\ &\quad S_0(t_i)^{\exp(x_i^T \hat{\beta}^{(m)} + v_i + \log(w_i^{(m)}))} \Big] \end{aligned} \tag{2.28}$$

This can be maximized through COXPH function, with offset term specified as $v_i + \log(w_i^{(m)})$.

To estimate the baseline survival $S_0(t_i)$, we order the event time and denote them as $t_{(1)} < t_{(2)} < t_{(3)} < \cdots < t_{(j)}$. Let the number of events be $d_{t_{(j)}}$, and the risk set be $R(t_{(j)})$ at time $t_{(j)}$. Again, we use the Breslow-type estimator with the estimated $\hat{b}^{(m)}$ and $\hat{\beta}^{(m)}$ in the m^{th} iteration, and $\hat{S}_0(t)$ is defined to be 0 when $t > t_{(k)}$ to assure that $\hat{S}_0(t)$ is a proper probability

$$\hat{S}_0^{(m)}(t_i) = \exp\left(- \sum_{j: t_{(j)} \le t} \frac{d_{t_{(j)}}}{\sum_{i \in R(t_{(j)})} w_i^{(m)} \exp(x_i^T \hat{\beta}^{(m)} + v_i)} \right) \tag{2.29}$$

The estimated survival function

$$\hat{S}^{(m)}(t_i|x_i, v_i) = \hat{S}_0(t_i)^{\exp(x_i^T \hat{\beta}^{(m)} + v_i)} \tag{2.30}$$

will be used to update E-step.

The steps of modified EM algorithm with offset terms are listed in Algorithm 2. The input values include N observations $O_i = (T_i, \delta_i, z_i, x_i, u_i, v_i)$, the maximum number of iteration times in the EM algorithm n_{max} and a convergence bound ϵ. The output values should be the estimation of $\hat{\Theta} = (\hat{b}, \hat{\beta}, \hat{S}_0(t))$, which can be used to calculate the survival function in population $\hat{S}_{pop}(t_i|z_i, x_i, u_i, v_i)$.

The initialization is as follows. $w_i^{(0)} = \delta_i$, $\hat{b}^{(0)}$ and $\hat{\beta}^{(0)}$ will be calculated using $w_i^{(0)}$ respectively through GLM function and COXPH function. Then, the initial values for $\hat{S}_0^{(0)}(t_i)$ and $\hat{S}^{(0)}(t_i|x_i, v_i)$ can be set. The E-step with those initial values is to calculate $w_i^{(m)}$ through Equation (2.26). In the M-step, $\hat{b}^{(m)}$, $\hat{\beta}^{(m)}$ and $\hat{S}^{(m)}(t_i|x_i, v_i)$ are updated using Equation (2.27) to Equation (2.30). The criteria to stop the iteration is 1) $|con| < \epsilon$ in Equation (2.17) or 2) reach the maximum number of iteration times n_{max}.

Algorithm 2 Modified EM algorithm with offset terms

input $O_i = (T_i, \delta_i, z_i, x_i, u_i, v_i)$, n_{max},ϵ, $(i = 1, 2, \cdots, N)$

output $\hat{\Theta} = (\hat{b}, \hat{\beta}, \hat{S}_0(t))$

1: Initialisation of $w_i^{(0)}$, $\hat{b}^{(0)}$, $\hat{\beta}^{(0)}$, $\hat{S}_0^{(0)}(t_i)$ and $\hat{S}^{(0)}(t_i|x_i, v_i)$

2: m=1

3: E-step: calculate $\pi(z_i, u_i)$ and $w_i^{(m)}$

4: M-step:

(I) Maximize $E(l_1(b; O))$ using the GLM function with quasi-binomial model to estimate $\hat{b}^{(m)}$

(II) Maximize $E(l_2(\beta, S_0(t); O))$ using COXPH function with offset $v_i + \log(w_i^{(m)})$ to estimate $\hat{\beta}^{(m)}$

(III) Estimate baseline survival function $\hat{S}_0^{(m)}(t_i)$

5: Check the iteration criteria:

If $|con| < \epsilon$ and iteration times $< n_{max}$, then $m = m + 1$ and update the parameters

else stop the iteration and report output

2.4 Variance estimation

The variance estimation in standard SMCURE is under bootstrap method, which draws bootstrap samples after the estimation of $\hat{\Theta}$ to represent the population, and calculates the variance between those bootstrap samples. In order to keep consistent with the standard SMCURE algorithm, we apply the bootstrap method for variance estimation in the modified EM algorithm, while adding discussion of situations regarding offset terms into the bootstrap procedure.

Based on the observations $O = \{O_1, O_2, \cdots, O_N\}$, $s^{(k)} = \{s_1^{(k)}, s_2^{(k)}, \cdots, s_n^{(k)} | k = 1, 2, \cdots, B\}$ denotes one of the B bootstrap samples with sample size N using SAMPLE function in R, via stratified simple random sampling with replacement. Note that the sampling method should be with replacement in order to use every bootstrap sample to represent the population. Let n_0 be the number of censored data in $s^{(k)}$ and n_1 be the number of uncensored data in $s^{(k)}$, and $n_0 + n_1 = N$. To keep the same censoring rate, stratified sampling is applied to the censored subset and the uncensored subset to make $\frac{n_0}{n_1}$ the same as the ratio of censored data to uncensored data in the original data set.

The variance is then calculated by the B parameters

$$Var(\hat{b}) = \frac{1}{B}\sum_{k=1}^{B}(\hat{b}_k - \bar{b})^2, \quad Var(\hat{\beta}) = \frac{1}{B}\sum_{k=1}^{B}(\hat{\beta}_k - \bar{\beta})^2 \tag{2.31}$$

Also, the standard deviations are

$$s(\hat{b}) = \sqrt{Var(\hat{b})}, \quad , s(\hat{\beta}) = \sqrt{Var(\hat{\beta})} \tag{2.32}$$

Algorithm 3 displays the steps of variance estimation process.

Algorithm 3 Bootstrap variance estimation

input $O_i = (T_i, \delta_i, z_i, x_i, u_i, v_i)$, B, $(i = 1, 2, \cdots, N)$

output $Var(\hat{b})$, $Var(\hat{\beta})$, $s(\hat{b})$, $s(\hat{\beta})$

1: $k = 1$

2: Split the input data into n_0 censored data and n_1 uncensored data.

3: Apply simple random sampling with replacement separately in censored data and uncensored data. Combine them together as a bootstrap sample.

4: In the bootstrap sample, apply the EM algorithm and acquire the estimators $\hat{b}_k$ and $\hat{\beta}_k$.

5: $k = k + 1$ until $k = B$.

6: Use $\hat{b}_1$, $\hat{b}_2$, $\cdots$,$\hat{b}_B$ and $\hat{\beta}_1$, $\hat{\beta}_2$, $\cdots$, $\hat{\beta}_B$ to compute the variance and standard deviation:

$$Var(\hat{b}) = \frac{1}{B}\sum_{k=1}^{B}(\hat{b}_k - \bar{b})^2, \quad Var(\hat{\beta}) = \frac{1}{B}\sum_{k=1}^{B}(\hat{\beta}_k - \bar{\beta})^2$$

$$s(\hat{b}) = \sqrt{Var(\hat{b})}, \quad , s(\hat{\beta}) = \sqrt{Var(\hat{\beta})}$$

Chapter 3

Numerical Experiment

This section mainly presents the numerical experiments results based on the simulated data and real data. Section 3.1 summarizes the simulation results to demonstrate the validity of the modified EM algorithm and Section 3.2 focuses on the real data analysis.

3.1 Simulation study

3.1.1 Overview

There are two simulation designs to evaluate the performances of the modified EM algorithm under following situations

- different censoring rates under three baseline distributions
- different combinations/formats of the offset terms under three baseline distributions

For each situation, the sample size is set as $N = 1000$ and the number of data sets is $n_{simu} = 500$.

For each simulation data set, we apply two algorithms listed below

1. the standard SMCURE without offset term. In this approach we will assume that the offset term is a regular covariate and we will fit the PHMC model directly.
2. the modified EM algorithm with a specified offset term

The results are reported by the average bias, the empirical variance and the average of the bootstrap variance. Let θ denote the unknown parameters i.e. b or β in this thesis. The average bias is defined as $\frac{1}{n_{simu}}(\hat{\theta} - \theta)$.

For the variance comparison, an average bootstrap variance in Equation (3.1) for each parameter is calculated, while an empirical variance in Equation (3.2) for each parameter is calculated only using the modified EM algorithm

$$\frac{1}{n_{simu}} \sum_{i=1}^{n_{simu}} Var_{boot}(\theta) \tag{3.1}$$

$$\frac{1}{1 - n_{simu}} \sum_{i=1}^{n_{simu}} (\hat{\theta} - \bar{\theta})^2 \tag{3.2}$$

When the bootstrap variance and the empirical variance are similar, it indicates that the bootstrap variance works well. In the thesis, we set the number of bootstrap samples as 100. For the purpose of convenience, "bias" means the average bias, "bootstrap" means the average bootstrap variance and "empirical" means the empirical variance in the Tables.

In all simulation settings, we include three covariates $x_i^T = (x_{1i}, x_{2i}, x_{3i})$, $(i = 1, 2, \cdots, N)$, where x_{1i} and x_{3i} respectively follow Binomial distributions with probabilities p_1 and p_2, and x_{2i} follows a standard Normal distribution. The incidence and latency are assumed to share the same covariates sets. The probability of a patient being uncured under the logistic regression is

$$\pi(x_i) = \frac{\exp(b_0 + b_1 x_{1i} + b_2 x_{2i} + b_3 x_{3i})}{1 + \exp(b_0 + b_1 x_{1i} + b_2 z_{2i} + b_3 x_{3i})} \tag{3.3}$$

as long as the parameter vector $b = (b_0, b_1, b_2, b_3)$ is given. The average cure rate can be calculated for the simulation data sets.

The survival function of the PH model can be written as $S(t_i|x_i) = S_0(t_i)^{\exp(x_i^T \beta)}$. Here, $x_i^T \beta = \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i}$, and the baseline survival distribution come from various distributions.Specifically, we consider the exponential distribution, the Weibull distribution and the log-Normal distribution. First, we generate the survival probability from a Uniform distribution $U(0, 1)$ and denote it as u_i^*. Second, the baseline

survival probability can be written as $S_0(t_i) = \exp(\frac{\log(u_i^*)}{\exp(x_i^T\beta)})$. Last, t_i^* can be solved from the formula that $1-S_0(t_i^*)$ is equal to a cumulative density function of a specified distribution mentioned above.

The censoring time C_i is generated from a Uniform distribution which reaches to different censoring rates. For each subject i, the observed time $T_i = \min\{t_i^*, C_i\}$. Based on the corresponding censoring indicator δ_i the survival time can be denoted by

$$T_i = \delta_i t_i^* + (1 - \delta_i)C_i \tag{3.4}$$

3.1.2 Simulation Setting 1: Different censoring rates

In the first simulation setting, x_1 and x_3 are randomly generated from Binomial distribution with probabilities as $p_1 = 0.5$ and $p_2 = 0.2$. The three baseline distributions are taking Weibull(5,5), log-Normal(2,0.1) and exponential(2). We set coefficient $b = (b_0, b_1, b_2, b_3) = (2, -1, 2, 1)$, and $\beta^T = (\beta_1, \beta_2, \beta_3) = (-2, 1, 1)$. The covariate x_3 is considered an offset term for both incidence and latency parts. Note that we will not estimate the related coefficient of x_3 in the modified EM algorithm. The average cure rates from three baseline distributions are around 26%, and some medium censoring rates (30% - 45%) are generated via changing the parameters in the Uniform distributions for the censored survival time. Table 3.1 shows the detailed of the average cure rates and censoring rates for each data set setting which are denoted as Weibull1, Log Normal1, exponentail1, Weibull2, Log Normal2, and exponentail2.

Table 3.1 Simulation data settings for different censoring rates

p_1	p_2	censored time	baseline distribution	cure rates	censoring rates	setting name
0.5	0.2	Uniform(2,50)	Weibull(5,5)	0.2654	0.3272	Weibull 1
			log-Normal(2,0.1)	0.2657	0.3728	log-Normal 1
			exponential(2)	0.2678	0.2979	exponential 1
		Uniform(2,15)	Weibull(5,5)	0.2669	0.4400	Weibull 2
		Uniform(2,25)	log-Normal(2,0.1)	0.2658	0.4667	log-Normal 2
		Uniform(2,20)	exponential(2)	0.2659	0.3132	exponential 2

The results of average bias and variance for Weibull 1 data set and Weibull 2 data set are in Table 3.2. The modified EM algorithm with offset x_3 has smaller average bias values for $\hat{b}_0$, $\hat{b}_1$, and $\hat{b}_2$ compared to the standard SMCURE without offset terms in Weibull 1 data set, and it also has smaller average bias values for $\hat{b}_0$, $\hat{b}_1$, $\hat{\beta}_1$, and $\hat{\beta}_2$ in Weibull 2 data set. For the average bootstrap variance, the modified EM algorithm obtains smaller values for $\hat{b}_0$, $\hat{b}_1$, $\hat{b}_2$, $\hat{\beta}_1$, and $\hat{\beta}_2$ compared to the standard SMCURE both in Weibull 1 and Weibull 2 data sets. The empirical variance from modified EM algorithm is similar to the average of the bootstrap variance.

Table 3.2 Simulation results for Weibull data set

algorithm	estimation	$b_0 = 2$	$b_1 = -1$	$b_2 = 2$	$b_3 = 1$	$\beta_1 = -2$	$\beta_2 = 1$	$\beta_3 = 1$
Weibull 1 data set								
SMCURE	bias	0.0446	-0.0089	0.0716	0.2217	0.0288	-0.0238	0.0076
	bootstrap	0.2072	0.3218	0.2269	6.7012	0.0590	0.0349	0.0597
modified EM algorithm	bias	0.0268	0.0037	0.0487		0.0330	-0.0306	
	bootstrap	0.0262	0.0487	0.0294		0.0097	0.0054	
	empirical	0.0313	0.0470	0.0313		0.0099	0.0056	
Weibull 2 data set								
SMCURE without offset	bias	0.0158	0.0120	-0.0020	-0.0120	0.0196	-0.0252	-0.0077
	bootstrap	0.0417	0.0771	0.0484	0.1427	0.0124	0.0069	0.0120
modified SMCURE with offset x_3	bias	0.0104	0.0117	-0.0030		0.0191	-0.0245	
	bootstrap	0.0397	0.0755	0.0456		0.0119	0.0064	
	empirical	0.0453	0.0706	0.0461		0.0107	0.0063	

The results of average bias and variance for log-Normal 1 data set and log-Normal 2 data set are in Table 3.3. The modified EM algorithm with offset x_3 produces smaller average bias values for all parameters in log-Normal 1 data set, and also has smaller average bias values for $\hat{b}_0$ and $\hat{\beta}_1$. The average bias for $\hat{\beta}_2$ are the same for both algorithms. For the average bootstrap variance, the modified EM algorithm with offset x_3 acquires smaller values in all parameters compared to the standard SMCURE algorithm. The empirical variance from modified EM algorithm is close to the average bootstrap variance.

Table 3.3 Simulation results for log-Normal data set

algorithm	estimation	$b_0 = 2$	$b_1 = -1$	$b_2 = 2$	$b_3 = 1$	$\beta_1 = -2$	$\beta_2 = 1$	$\beta_3 = 1$
log-Normal 1 data set								
SMCURE	bias	0.0296	-0.0132	0.0159	0.0278	0.0380	-0.0410	-0.0217
	bootstrap	0.0312	0.0542	0.0348	0.0965	0.0108	0.0062	0.0106
modified EM algorithm	bias	0.0280	-0.0101	0.0104		0.0345	-0.0371	
	bootstrap	0.0301	0.0529	0.0328		0.0104	0.0058	
	empirical	0.0352	0.0535	0.0317		0.0111	0.0062	
log-Normal 2 data set								
SMCURE	bias	0.0114	0.0136	-0.0015	0.0129	0.0342	-0.0300	-0.0050
	bootstrap	0.0397	0.0726	0.0451	0.1347	0.0132	0.0075	0.0128
modified EM algorithm	bias	0.0074	0.0159	-0.0067		0.0340	-0.0300	
	bootstrap	0.0385	0.0713	0.0425		0.0124	0.0068	
	empirical	0.0414	0.0683	0.0414		0.0122	0.0076	

The results of average bias and variance for exponential 1 data set and exponential 2 data set are in Table 3.4. The modified EM algorithm with offset x_3 has smaller average bias values for $\hat{\beta}_1$ and $\hat{\beta}_2$ compared to the standard SMCURE algorithm in exponential 1 data set. In exponential 2 data set, the modified EM algorithm obtains smaller average bias values for $\hat{b}_1$, $\hat{b}_2$, $\hat{\beta}_1$ and $\hat{\beta}_2$. For the average bootstrap variance, the modified EM algorithm is producing smaller values for all parameters compared to the standard SMCURE algorithm. The empirical variance from modified EM algorithm is very similar to the average bootstrap variance.

Table 3.4 Simulation results for exponential data set

algorithm	estimation	$b_0 = 2$	$b_1 = -1$	$b_2 = 2$	$b_3 = 1$	$\beta_1 = -2$	$\beta_2 = 1$	$\beta_3 = 1$
exponential 1 data set								
SMCURE	bias	-0.0071	0.0052	-0.0034	0.0171	0.0438	-0.0490	-0.0180
	bootstrap	0.0245	0.0471	0.0300	0.0827	0.0099	0.0057	0.0095
modified EM algorithm	bias	-0.0084	0.0073	-0.0076		0.0408	-0.0458	
	bootstrap	0.0235	0.0461	0.0285		0.0093	0.0052	
	empirical	0.0309	0.0472	0.0283		0.0086	0.0055	
exponential 2 data set								
SMCURE	bias	0.0015	0.0581	-0.0509	-0.0292	0.0275	-0.0276	-0.0117
	bootstrap	0.0397	0.0726	0.0451	0.1347	0.0132	0.0075	0.0128
modified EM algorithm	bias	-0.0045	0.0555	-0.0483		0.0266	-0.0263	
	bootstrap	0.0255	0.0619	0.0357		0.0098	0.0056	
	empirical	0.0346	0.0633	0.0368		0.0095	0.0057	

The above comparison between the standard SMCURE algorithm and the modified EM algorithm indicates that the modified EM algorithm has a better performance in estimation when considering an offset term in data among different censoring rates. Also, the variance results show that the bootstrap method for variance estimation is feasible under such conditions.

The average iteration times are recorded for the modified EM algorithm, which are listed in Table 3.5. From the table, we found that when the censoring rates increase, the average iteration times increase. This is consistent with our hypothesis since the high the censoring the less information about the true event time.

Table 3.5 Average iteration times using modified EM algorithm

name	Weibull 1	Weibull 2	log-Normal 1	log-Normal 2	exponential 1	exponential 2
iteration times	5.042	12.144	6.202	10.564	5.012	12.026

3.1.3 Simulation Setting 2: Different variables for the offset term

To test different covariates, categorical or continuous or the combination in the incidence and latency components, as the offset terms, the simulation design includes:

- (1) x_3 as an offset term both in incidence part and in latency part, and the parameters are set as $b = (2, -1, 2, 1)$ and $\beta^T = (-2, 1, 1)$;
- (2) x_2 as an offset term both in incidence part and in latency part, and the parameters are set as $b = (2, -1, 1, 2)$ and $\beta^T = (-2, 1, 2)$;
- (3) x_2 is an incidence offset term, and x_3 is a latency offset term, with the parameters $b = (2, -1, 2, 2)$ and $\beta^T = (-2, 2, 1)$;
- (4) x_2 is a latency offset term and x_3 is an incidence offset term, with the parameters $b = (2, -1, 2, 1)$ and $\beta^T = (-2, 1, 2)$.

Note that the coefficient of an offset term is considered as 1 and will not be estimated. The specific settings are illustrated in Table 3.6, with the average of cure rates around 10% - 20% and censoring rates around 10% - 30%.

Table 3.6 The cure rates and censoring rates of four types of parameter settings

setting name	baseline distribution	cure rate	censoring rate
(1)	Weibull(5,5)	0.2223	0.2707
	log-Normal(2,0.1)	0.2221	0.3177
	exponential(2)	0.2218	0.2315
(2)	Weibull(5,5)	0.1211	0.1710
	log-Normal(2,0.1)	0.1216	0.2265
	exponential(2)	0.1211	0.1335
(3)	Weibull(5,5)	0.2112	0.1202
	log-Normal(2,0.1)	0.1210	0.2644
	exponential(2)	0.1202	0.1759
(4)	Weibull(5,5)	0.2213	0.2608
	log-Normal(2,0.1)	0.2216	0.3125
	exponential(2)	0.2221	0.2292

In these four settingsresults from the standard SMCURE without offset terms and the modified EM algorithm with different offset terms are compared to verify the validity of the modified EM algorithm. We only calculate the empirical variance from the modified EM algorithm to evaluate the performance of the bootstrap method.

The simulation results of average bias and variance of situation (1) are in Table 3.7.

Table 3.7 Simulation results of bias and variance with x_3 both as the incidence offset and the latency offset term

baseline type	algorithm	estimation	$b_0 = 2$	$b_1 = -1$	$b_2 = 2$	$b_3 = 1$	$\beta_1 = -2$	$\beta_2 = 1$	$\beta_3 = 1$
Weibull(5,5)	SMCURE	bias	0.0198	-0.0023	0.0359	0.0071	0.0019	-0.0030	-0.0031
		bootstrap	0.0399	0.0479	0.0277	0.0478	0.0095	0.0028	0.0072
	modified EM algorithm	bias	0.0184	0.0003	0.0313		0.0026	-0.0032	
		bootstrap	0.0382	0.0464	0.0257		0.0090	0.0026	
		empirical	0.0404	0.0494	0.0254		0.0090	0.0025	
log-Normal(2,0.1)	SMCURE	bias	0.0213	-0.0187	0.0240	0.0082	0.0019	-0.0083	-0.0101
		bootstrap	0.0448	0.0516	0.0305	0.0509	0.0104	0.0030	0.0078
	modified EM algorithm	bias	0.0190	-0.0146	0.0190		0.0007	-0.0074	
		bootstrap	0.0415	0.0498	0.0277		0.0099	0.0028	
		empirical	0.0400	0.0492	0.0272		0.0095	0.0028	
exponential(2)	SMCURE	bias	0.0243	-0.0209	0.0139	0.0063	0.0064	-0.0031	-0.0079
		bootstrap	0.0366	0.0451	0.0258	0.0446	0.0091	0.0027	0.0068
	modified EM algorithm	bias	0.0222	-0.0185	0.0093		0.0055	-0.0023	
		bootstrap	0.0358	0.0434	0.0241		0.0086	0.0025	
		empirical	0.0309	0.0407	0.0208		0.0088	0.0026	

From the table, the values of average bias and average bootstrap variance among three baseline types from modified EM algorithm with x_3 as both the incidence and latency offset are generally smaller than those from the standard SMCURE. Based on the modified EM algorithm, the empirical variance is generally close to but slightly smaller than the average bootstrap variance.

The simulation results of average bias and variance of situation (2) are in Table 3.8. In this case, because x_2 is considered an offset both in incidence and latency, the coefficient for it is not estimated in the modified EM algorithm. From the Table 3.8, almost all of the values of average bias from the modified EM algorithm are smaller than those from the standard SMCURE among three baseline distributions, indicating a better performance of the modified EM algorithm in terms of the specified offset term. The values of average bootstrap variance from the modified EM algorithm are smaller for $\hat{b}_1$, $\hat{\beta}_1$ and $\hat{\beta}_3$ compared to the standard SMCURE algorithm. The values of empirical variance are similar to those of average bootstrap variance.

Table 3.8 Simulation results of bias and variance with x_2 both as the incidence offset and the latency offset term

baseline type	algorithm	estimation	$b_0 = 2$	$b_1 = -1$	$b_2 = 1$	$b_3 = 2$	$\beta_1 = -2$	$\beta_2 = 1$	$\beta_3 = 2$
Weibull(5,5)	SMCURE	bias	0.0098	0.0038	0.0019	0.0176	0.0061	-0.0083	-0.0202
		bootstrap	0.0389	0.0632	0.0196	0.0740	0.0083	0.0021	0.0091
	modified EM algorithm	bias	0.0058	0.0055		0.0140	0.0004		-0.0139
		bootstrap	0.0413	0.0623		0.0706	0.0064		0.0017
		empirical	0.0396	0.0553		0.0662	0.0066		0.0077
log-Normal(2,0.1)	SMCURE	bias	0.0108	-0.0027	0.0012	0.0250	0.0131	-0.0115	-0.0185
		bootstrap	0.0389	0.0632	0.0196	0.0740	0.0083	0.0021	0.0091
	modified EM algorithm	bias	0.0071	-0.0011		0.0213	0.0044		-0.0090
		bootstrap	0.0440	0.0663		0.0783	0.0069		0.0019
		empirical	0.0468	0.0638		0.0728	0.0070		0.0089
exponential(2)	SMCURE	bias	0.0241	-0.0047	-0.0024	-0.0008	0.0048	-0.0054	-0.0150
		bootstrap	0.0335	0.0607	0.0188	0.0698	0.0079	0.0021	0.0086
	modified EM algorithm	bias	0.0223	-0.0038		-0.0033	0.0010		-0.0108
		bootstrap	0.0377	0.0591		0.0689	0.0061		0.0016
		empirical	0.0363	0.0581		0.0615	0.0066		0.0068

The simulation results of average bias and variance in situation (3) are in Table 3.9. The coefficients for the incidence offset x_2 and latency offset x_3 do not need estimation. From the table, the values of average bias among three baseline distributions from the modified EM algorithm is generally smaller than those from the standard SMCURE algorithm without offset terms. For the average bootstrap variance, the modified EM algorithm produces smaller values for $\hat{b}_1$, $\hat{b}_3$, $\hat{\beta}_1$ and $\hat{\beta}_2$, compared to the standard SMCURE algorithm. In this case, the values of empirical variance and average bootstrap variance close to each other.

Table 3.9 Simulation results of bias and variance with x_2 as the incidence offset and x_3 as the latency offset

baseline type	algorithm	estimation	$b_0 = 2$	$b_1 = -1$	$b_2 = 1$	$b_3 = 2$	$\beta_1 = -2$	$\beta_2 = 1$	$\beta_3 = 2$
Weibull(5,5)	SMCURE	bias	0.0242	-0.0154	0.0097	0.0426	0.0724	-0.0696	-0.0360
		bootstrap	0.0432	0.0758	0.0292	0.1011	0.0084	0.0046	0.0068
	modified EM algorithm	bias	0.0159	-0.0094		0.0343	0.0631	-0.0612	
		bootstrap	0.0466	0.0710		0.0924	0.0080	0.0041	
		empirical	0.0462	0.0669		0.0919	0.0071	0.0043	
log-Normal(2,0.1)	SMCURE	bias	0.0213	-0.0114	-0.0069	0.0515	0.0598	-0.0658	-0.0275
		bootstrap	0.0504	0.0833	0.0316	0.1033	0.0092	0.0050	0.0074
	modified EM algorithm	bias	0.0191	-0.0112		0.0472	0.0539	-0.0597	
		bootstrap	0.0521	0.0788		0.0970	0.0087	0.0045	
		empirical	0.0523	0.0768		0.1011	0.0087	0.0042	
exponential(2)	SMCURE	bias	0.0302	-0.0099	-0.0132	0.0293	0.0689	-0.0643	-0.0380
		bootstrap	0.0378	0.0733	0.0276	0.0955	0.0082	0.0045	0.0066
	modified EM algorithm	bias	0.0332	-0.0138		0.0276	0.0601	-0.0555	
		bootstrap	0.0460	0.0704		0.0875	0.0078	0.0040	
		empirical	0.0408	0.0631		0.0804	0.0078	0.0041	

The simulation results of average bias and variance in situation (4) are in Table 3.10. The coefficients for the incidence offset x_3 and latency offset x_2 do not need estimation. According to the table, the values of average bias under three different baseline distributions from the modified EM algorithm are smaller than those form the standard SMCURE. The values of average bootstrap variance for $\hat{b}_1$, $\hat{b}_2$, $\hat{\beta}_1$ and $\hat{\beta}_3$ from the modified EM algorithm are also smaller. Similarly, the values of empirical variance and average bootstrap variance are relatively close to each other.

Table 3.10 Simulation results of bias and variance with x_3 as the incidence offset and x_2 as the latency offset

baseline type	algorithm	estimation	$b_0 = 2$	$b_1 = -1$	$b_2 = 1$	$b_3 = 2$	$\beta_1 = -2$	$\beta_2 = 1$	$\beta_3 = 2$
Weibull(5,5)	SMCURE	bias	0.0269	-0.0098	0.0220	-0.0036	-0.0029	-0.0008	0.0036
		bootstrap	0.0394	0.0466	0.0262	0.0461	0.0093	0.0028	0.0101
	modified EM algorithm	bias	0.0197	-0.0080	0.0185		-0.0028		0.0036
		bootstrap	0.0319	0.0458	0.0247		0.0079		0.0085
		empirical	0.0354	0.0426	0.0251		0.0082		0.0079
log-Normal(2,0.1)	SMCURE	bias	0.0228	-0.0048	0.0285	0.0091	-0.0048	0.0014	-0.0062
		bootstrap	0.0432	0.0497	0.0291	0.0503	0.0101	0.0030	0.0109
	modified EM algorithm	bias	0.0217	-0.0018	0.0231		-0.0028		-0.0082
		bootstrap	0.0355	0.0487	0.0265		0.0085		0.0091
		empirical	0.0388	0.0461	0.0242		0.0086		0.0094
exponential(2)	SMCURE	bias	0.0266	-0.0071	0.0288	-0.0085	-0.0011	-0.0019	-0.0068
		bootstrap	0.0361	0.0444	0.0248	0.0449	0.0092	0.0027	0.0097
	modified EM algorithm	bias	0.0185	-0.0060	0.0266		-0.0019		-0.0061
		bootstrap	0.0290	0.0430	0.0232		0.0076		0.0082
		empirical	0.0336	0.0387	0.0224		0.0078		0.0088

The results based on four different offset designs indicate a better estimation performance for the modified EM algorithm among three different baseline distributions than the standard SMCURE algorithm. The bootstrap method is feasible for the variance estimation, with smaller values from the modified EM algorithm, and also produces similar results to the empirical variance.

Additionally, the average iteration times for the four situations from the modified EM algorithm are in Table 3.11. It is clear that for each baseline distribution in these four settings, the average iteration times are remaining at some close numbers like about 5 times for Weibull distribution, and far smaller than the convergence criteria.

Table 3.11 Average iteration times using modified EM algorithm

	baseline type		
setting name	Weibull	log-Normal	exponential
(1)	5.082	6.194	4.44
(2)	5.286	6.41	4.968
(3)	5.02	6.164	4.256
(4)	5.51	7.01	5.088

3.2 Real data study

The bone marrow transplant (BMT) data from article Copelan et al. (1991)) was acquired in R package GEECURE (Niu et al. (2018)), which focused on the effect of $BuCy_2$ as a regimen in the bone marrow transplantation for acute myelocytic leukemia (AML) patients, and examined the risk factors for the relapse and long-term leukemia-free survival (LFS) of leukemia patient. There is a total of $n = 137$ patients, and the main outcome is the disease-free survival time (time to relapse, death or end of study) with an overall censoring rate as 40.8759%. The Kaplan-meier (KM) survival curve with a 95% confidence interval for the overall survival time is illustrated in Figure 3.1.

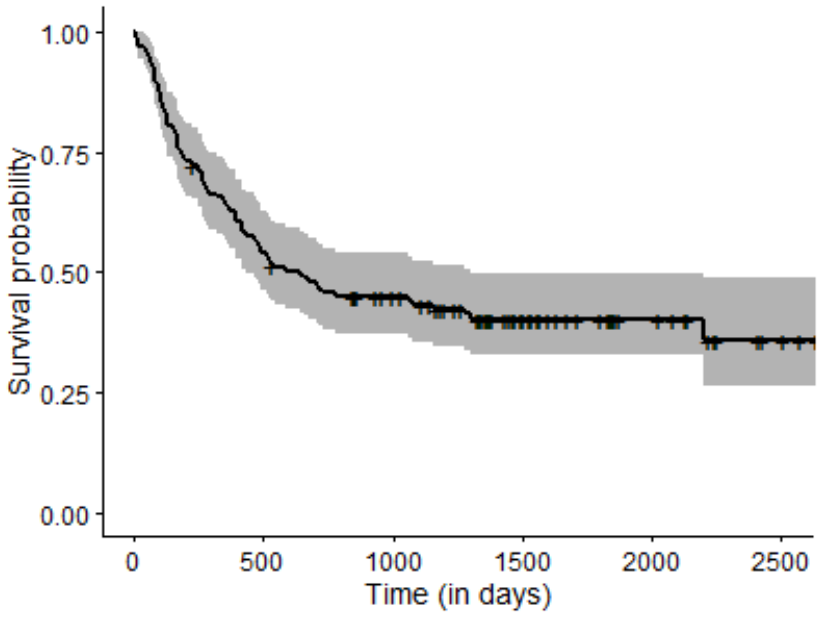

Figure 3.1 Kaplan-meier survival curve for BMT data

It is apparent that the survival probability does not decrease to 0, and indicates that there might be cured patients in this data set. Therefore, we consider the PHMC model. The potential risk factors considered in this illustration are listed in Table 3.12.

Table 3.12 Covariates information in BMT data

covariate	description	possible values	range or frequency
paage	patient age	numeric	(7,52)
doage	donor age	numeric	(2,56)
pasex	patient gender	1 - Male, 0 - Female	80 males and 57 females
dosex	donor gender	1 - Male, 0 - Female	88 males and 49 females
pa.CMV	patient CMV status	1 - positive, 0 - negative	68 positive and 69 negative
do.CMV	donor CMV status	1 - positive, 0 - negative	58 positive and 79 negative
wait	waiting time to transplant in days	numeric	(24,2616)
FAB	French-American-British classification	1 - Grade 4 or 5 and AML 0 - otherwise	45 patients 92 patients
MTX	MTX used as a GVHP	1 - yes 0 - no	40 patients 97 patients

Note that CMV status refers to the cytomegalovirus testing result, GVHP means the Graft-Versus-Host-Prophylactic and MTX refers to methotrexate. We further

eaxamine the Kaplan-meier curves with respect to MTX status in Figure 3.2, which also indicates the potential cure for patients with and without methotrexate.

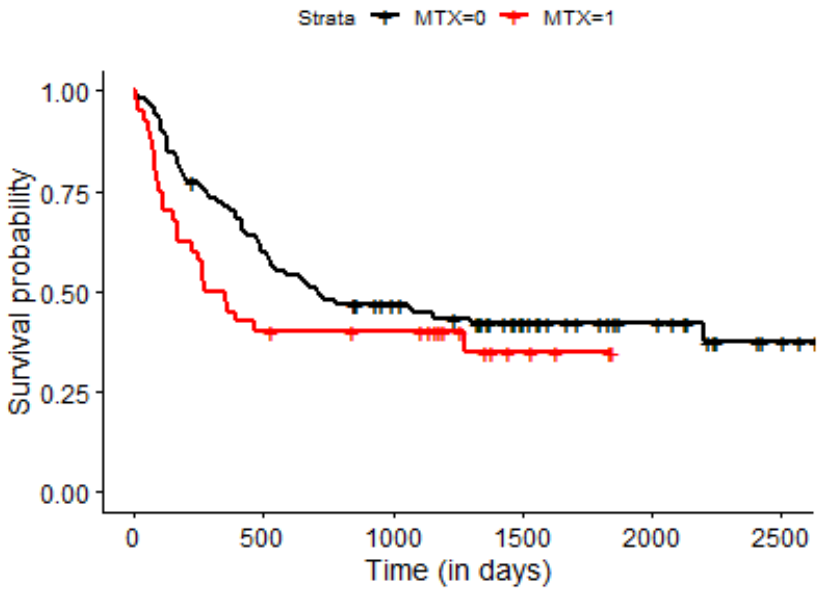

Figure 3.2 Kaplan-meier survival curves for BMT data in groups with and without MTX

Therefore, we apply the PHMC model to BMT data set and all the risk factors listed in Table 3.12 are taken into consideration in both incidence and latency part. The results of estimated coefficient and related variance and p-values from SMCURE without offset terms are listed in Table 3.13. The p-value for 0 refers to test whether the estimate is significantly different from 0, and the p-value for 1 refers to test whether the estimate is significantly different from 1. The number of bootstrap samples used is $B = 100$.

Table 3.13 Results of SMCURE for all risk factors

component	covariate	estimate	variance	p-value for 0	p-value for 1
incidence	intercept	0.2742	1.1983	0.8022	0.6472
	paage	-0.0480	0.0039	0.4434	<0.0001
	doage	0.0513	0.0032	0.3651	<0.0001
	pasex	-0.1515	0.4115	0.8133	0.1364
	dosex	-0.1954	0.3607	0.7449	0.1060
	pa.CMV	0.6505	0.5948	0.3990	0.7260
	do.CMV	-0.1367	0.5332	0.8515	0.1065
	wait	0.0013	<0.0001	0.7438	<0.0001
	FAB	0.6620	0.5289	0.3627	0.7431
	MTX	-0.4818	1.2645	0.6683	0.5860
latency	paage	0.0171	0.0016	0.6695	<0.0001
	doage	0.0118	0.0015	0.7603	<0.0001
	pasex	-0.2190	0.2005	0.6248	0.0115
	dosex	0.1871	0.2134	0.6854	0.0475
	pa.CMV	-0.6972	0.2418	0.1562	0.0002
	do.CMV	0.2211	0.2104	0.6298	0.0418
	wait	-0.0005	<0.0001	0.5984	<0.0001
	FAB	0.7695	0.2862	0.1503	0.6735
	MTX	1.4739	0.3603	0.0141	0.4160

According to the table, there is no covariate for the incidence part significantly different from 0, while patients with/without methotrexate plays an significant role in survival probability (p-value = 0.0141) at the significance level of 0.05. Patients with methotrexate have a higher risk to die. In order to examine the potential offset term, we further set the covariates of which the estimated coefficient is close to 1 as an offset term. For illustration, we select FAB as an offset term in the latency part. Then, we apply the standard SMCURE and the modified EM algorithm to fit the BMT data set. For the incidence, we include the patients CMV status since it has the lowest p-value among others, and consider the MTX classification in the latency part, when FAB is a latency offset term because the p-value 0.6735 indicates that FAB is not significantly different from 1. Results are listed in Table 3.14. With the FAB as the offset term, none of the risk factors are significant in the modified algorithm

comparing to the PHMC model without offset term.

Table 3.14 Results from two algorithms for BMT data

algorithm	component	covariate	estimate	variance	p-value
SMCURE	incidence	intercept	0.6513	0.0803	0.0215
		pa.CMV	0.2210	0.1744	0.5968
	latency	FAB	0.7583	0.1424	0.0445
		MTX	1.3308	0.1352	0.0003
modified EM algorithm	incidence	intercept	0.6929	0.0994	0.0280
		pa.CMV	0.2412	0.2548	0.6327
	latency	FAB			
		MTX	1.4324	0.2240	0.0025

We further demonstrated the estimated survival curves from the modified EM algorithm for patient with/without MTX or not given patients has CMV in Figure 3.3. From the figure, patients using MTX have a smaller survival probability to relapse, death or end of study, compared to the patients not taking MTX. As indicated in the table, the difference is significant.

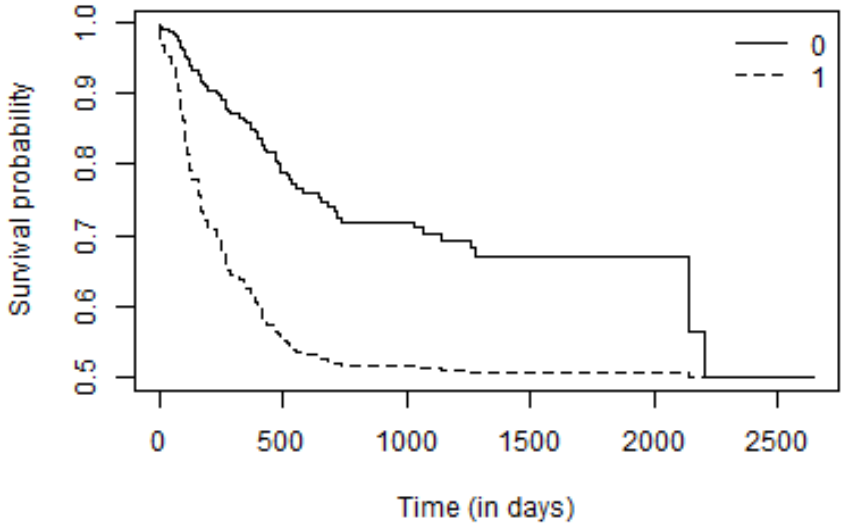

Figure 3.3 Estimated survival curves for patients using MTX or not

Chapter 4

Summary and Future work

With the medical advancement, the mixture cure model is becoming increasingly popular in handling the disease with a potential cure proportion. The R package SMCURE was proposed to estimate the PH mixture cure model and AFT mixture cure model based on the EM algorithm, while it did not adjust the offset terms properly. Therefore, in this thesis we considered the PH mixture cure model with offset. First, the PH mixture cure model examined in this thesis could account for offset terms separately in the incidence part and the latency part. Second, we modified the EM algorithm for the PH mixture cure model. Third, we designed a comprehensive simulation studies. Based on the simulation results, the modified EM algorithm performs well under different censoring rates, different baseline distributions and different combinations of offset terms. In analysis to the bone marrow transplant data and cooperative oncology group data, the modified EM algorithm could handle a separate offset term in the latency part and achieved an estimated survival curve.

The thesis only considers the PH model for the latency part, and future work could include modification to the AFT mixture cure model which assumes the AFT model for the latency part. The bootstrap method is applied to the variance estimation. Specifically, the stratified simple random sampling was applied to the censored data and uncensored data separately. Due to the resampling, the variance estimation is extremely time-consuming. Other variance estimation method should be investigated in the future to improve the process. For example, the perturbation method (Gu et al. (2020)) could be considered here. Through utilizing the simple perturb from

the exponential distribution with the mean and variance at 1, the approximation of the variance could be improved. The difficulty of variance estimation comes from the non-parametric estimation of the baseline survival function. Recently, some spline approaches were used to the mixture cure model, it could be extended to the mixture cure model with the offset terms.

Bibliography

Beran, R. (1981). Nonparametric regression with randomly censored survival data.

Berkson, J. and R. P. Gage (1952). Survival curve for cancer patients following treatment. *Journal of the American Statistical Association 47*(259), 501–515.

Boag, J. W. (1949). 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.

Breslow, N. (1974). Covariance analysis of censored survival data. *Biometrics*, 89–99.

Cai, C., Y. Zou, Y. Peng, and J. Zhang (2012). smcure: An r-package for estimating semiparametric mixture cure models. *Computer methods and programs in biomedicine 108*(3), 1255–1260.

Copelan, E. A., J. C. Biggs, J. M. Thompson, P. Crilley, J. Szer, J. P. Klein, N. Kapoor, B. R. Avalos, I. Cunningham, and K. Atkinson (1991). Treatment for acute myelocytic leukemia with allogeneic bone marrow transplantation following preparation with bucy2.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological) 34*(2), 187–202.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological) 39*(1), 1–22.

Farewell, V. T. (1982). The use of mixture models for the analysis of survival data with long-term survivors. *Biometrics*, 1041–1046.

Farewell, V. T. and R. L. Prentice (1977). A study of distributional shape in life testing. *Technometrics 19*(1), 69–75.

Ghitany, M., R. A. Maller, and S. Zhou (1994). Exponential mixture models with

long-term survivors and covariates. *Journal of multivariate Analysis 49*(2), 218–241.

Gu, E., J. Zhang, W. Lu, L. Wang, and F. Felizzi (2020). Semiparametric estimation of the cure fraction in population-based cancer survival analysis. *Statistics in Medicine 39*(26), 3787–3805.

Kalbfleisch, J. D. and R. L. Prentice (1973). Marginal likelihoods based on cox's regression and life model. *Biometrika 60*(2), 267–278.

Kleinbaum, D. G., M. Klein, et al. (2012). *Survival analysis: a self-learning text*, Volume 3. Springer.

Kuk, A. Y. and C.-H. Chen (1992). A mixture model combining logistic regression with proportional hazards regression. *Biometrika 79*(3), 531–541.

Lam, K., D. Y. Fong, and O. Tang (2005). Estimating the proportion of cured patients in a censored sample. *Statistics in medicine 24*(12), 1865–1879.

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

López-Cheda, A., R. Cao, M. A. Jácome, and I. Van Keilegom (2017). Nonparametric incidence estimation and bootstrap bandwidth selection in mixture cure models. *Computational Statistics & Data Analysis 105*, 144–165.

Lu, W. (2008). Maximum likelihood estimation in the proportional hazards cure model. *Annals of the Institute of Statistical Mathematics 60*(3), 545–574.

Niu, Y., X. Wang, and Y. Peng (2018). geecure: An r-package for marginal proportional hazards mixture cure models. *Computer methods and programs in biomedicine 161*, 115–124.

Peng, Y., K. B. Dear, and J. Denham (1998). A generalized f mixture model for cure rate estimation. *Statistics in medicine 17*(8), 813–830.

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

Wang, L., P. Du, and H. Liang (2012). Two-component mixture cure rate model with spline estimated nonparametric components. *Biometrics 68*(3), 726–735.

Yamaguchi, K. (1992). 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*(418), 284–292.

Appendix A

R code for the modified EM algorithm

```
#####2022.7.18

library("survival")
# modified SMCURE with separate offsets and bootstrap
   variance
# variance bootstrap samples are separately sampling from
   censored and uncensored data
## offset in incidence part and offset in latency part
##############################
##############################

## smsurv
smsurv<-function(Time, Status, X, beta, latency.offsetvar=
    NULL, w, model=c("ph","aft")){

  death_point <- sort(unique(subset(Time, Status==1)))

  #add offset term here NEWLY!
  if(model == 'ph'){
```

```
  if(!is.null(latency.offsetvar)) {
    coxexp <- exp( beta%*%t(X[,-1]) + 1%*%t(latency.
       offsetvar)  )
  }else {
    coxexp <- exp((beta)%*%t(X[,-1]))
  }
}else {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
em<-function(Time, Status, X, Z, incidence.offsetvar =
   NULL, latency.offsetvar = NULL,
              b, beta, model = c("ph","aft"), link, emmax =
                  100, eps=1e-7){
  w <- Status
  n <- length(Status)

  #####here should be part of survival for ph and AFT
  if(model == "ph") {
    s <- smsurv(Time, Status, X, beta, latency.offsetvar,
       w, model="ph")$survival
  }


  convergence<- 1000
  i <-1
```

```
iternum<-1

# prob of being uncured
while(convergence>eps & i<emmax){
  if(!is.null(incidence.offsetvar)){
    uncureprob <- matrix(exp(b%*%t(Z) + 1%*%t(incidence.
       offsetvar))/(1+exp(b%*%t(Z) + 1%*%t(incidence.
       offsetvar))),ncol=1)  #piz
  }else{
    uncureprob <- matrix(exp(b%*%t(Z))/(1+exp(b%*%t(Z)))
       ,ncol=1)
  }

  #survival functions
  if(model == "ph"){
    if(!is.null(latency.offsetvar)) {
      survival<-drop(s^(exp(beta%*%t(X[,-1]) + 1%*%t(
         latency.offsetvar))))
    }else {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

#update_cureb <- logistfit$coef
if(!is.null(incidence.offsetvar)) {
  update_cureb <- as.numeric(eval(parse(text = paste("
     glm", "(", "w~Z[,-1]+offset(incidence.offsetvar)
     ",",family = quasibinomial(link='", "'",")",")",
     sep = ""))))$coef)
  logistfit<- eval(parse(text = paste("glm", "(", "w~Z
     [,-1]+offset(incidence.offsetvar)",
              ",family = quasibinomial(link='",
            link, "'",")",")",sep = "")))
}else{
  update_cureb <- as.numeric(eval(parse(text = paste("
     glm", "(", "w~Z[,-1]",",family = quasibinomial(
     link='", link, "'",")",")",sep = ""))))$coef)
  logistfit<- eval(parse(text = paste("glm", "(", "w~Z
     [,-1]",",family = quasibinomial(link='", link,
     "'",")",")",sep = "")))
}

#update_beta and update_s
if(model == "ph") {
  if(!is.null(latency.offsetvar)) {
    update_beta <- coxph(Surv(Time, Status)~X[,-1]+
      offset(latency.offsetvar+log(w)),
                 subset=w!=0, method="breslow")$coef
```

```
  }else{
    update_beta <- coxph(Surv(Time, Status)~X[,-1]+
      offset(log(w)),
                  subset=w!=0, method="breslow")$coef
  }
  update_s <-smsurv(Time, Status, X, beta, latency.
    offsetvar, w, model="ph")$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))$par
  update_s <- smsurv(error,Status,X,beta,latency.
    offsetvar,w,model="aft")$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

#iteration times
iternum<-i

if(!is.null(incidence.offsetvar)){
  uncureprob <- matrix(exp(b%*%t(Z) + 1%*%t(incidence.
    offsetvar))/(1+exp(b%*%t(Z) + 1%*%t(incidence.
    offsetvar))),ncol=1)  #piz
```

```
    }else{
      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, iternum=
              iternum)
}

####################
####others for aft and print
#smrank
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
  }
```

```
#########################
#########################
# main body

sm5<-function(formula,cureform, incidence.offset=NULL,
   latency.offset=NULL,
              data,na.action=na.omit, model=c("ph","aft"),
              link="logit",Var=TRUE, emmax=100,eps=1e-7,
                 nboot=100){
  fit<-numeric()

  # check main body code
  call <- match.call()
  # model <- match.arg(model)

  ## data preparation
  ### normal covariates
  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)

  ### offset preparation
  if(!is.null(incidence.offset)) {
```

```
  inci.vars <- all.vars(incidence.offset)
  incidence.offsetvar <- as.matrix(data[,inci.vars])
  colnames(incidence.offsetvar)<-inci.vars}else {
     incidence.offsetvar <- NULL}

if(!is.null(latency.offset)){
  laten.vars<-all.vars(latency.offset)
  latency.offsetvar <- as.matrix(data[,laten.vars])
  colnames(latency.offsetvar)<-laten.vars}else {latency.
     offsetvar <- NULL}

### reframe the data input
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)}
```

```
incidence.nm<-colnames(incidence.offsetvar)
nincidence.off<-ncol(incidence.offsetvar)
latency.nm<-colnames(latency.offsetvar)
nlatency.off<-ncol(latency.offsetvar)

## initial value
w <- Status
if(!is.null(incidence.offsetvar)) {
  b <- eval(parse(text = paste("glm", "(",
                               "w~Z[,-1]+offset(
                                  incidence.offsetvar)",
                               ",family = quasibinomial(
                                  link='", link,
                               "'",")",")",sep = "")))
                                  $coef
}else{b <- eval(parse(text = paste("glm", "(", "w~Z
   [,-1]",",family = quasibinomial(link='", link, "'",")
   ",")",sep = "")))$coef}

if(model=="ph") {
  if(!is.null(latency.offsetvar)) {
    beta <- coxph(Surv(Time, Status)~X[,-1]+offset(log(w
       )+latency.offsetvar),
                  subset= w!=0, method="breslow")$coef
  }else{beta <- coxph(Surv(Time, Status)~X[,-1]+offset(
     log(w)), subset= w!=0, method="breslow")$coef}
}
```

```
if(model=="aft") beta <- survreg(Surv(Time,Status)~X
   [,-1])$coef

### do EM
emfit<-em(Time, Status, X,Z, incidence.offsetvar,
   latency.offsetvar,
          b, beta, model, link, emmax, eps)
b<-emfit$b
uncuredprob<-emfit$Ucureprob
beta<-emfit$latencyfit
s<-emfit$Survival
logistfit<-emfit$logistfit
iternum<-emfit$iternum

############################## bootstrap variance
   ##############################

if (Var) {
  if (model == "ph"){
    b_boot<-matrix(rep(0,nboot*nb), nrow = nboot)
    beta_boot<-matrix(rep(0,nboot*nbeta), nrow = nboot)
  }#if model = "aft"

  if(!is.null(incidence.offsetvar)){
    if(!is.null(latency.offsetvar)){
      # inci off+laten off
      # sampling according to censored and uncensored
```

```
   data
tempdata<-cbind(Time, Status, X, Z, incidence.
   offsetvar, latency.offsetvar)
data1<-subset(tempdata, Status==1) #uncensored
data0<-subset(tempdata, Status==0) #censored

ntemp<-nrow(tempdata)
n1<-nrow(data1)
n0<-nrow(data0)

i<-1
while(i <= nboot){

  #id<-sample(1:ntemp, size = ntemp, replace =
     TRUE)
  id1<-sample(1:n1, size = n1, replace = TRUE) #
     sampling from uncensored
  id0<-sample(1:n0, size = n0, replace = TRUE) #
     sampling from censored

  bootdata<-rbind(data1[id1,], data0[id0,])
  bootZ<-bootdata[,bnm]
  nbdata<-length(bootdata[,1])

  if(model == "ph"){
    bootX<-as.matrix( cbind( rep(1,nbdata) ,
       bootdata[,betanm]) )
```

```
    }
    # offset
    boot.inci.offsetvar<-bootdata[,incidence.nm]
    boot.laten.offsetvar<-bootdata[,latency.nm]

    bootfit<-em(bootdata[,1], bootdata[,2], bootX,
       bootZ,
                incidence.offsetvar = boot.inci.
                   offsetvar,
                latency.offsetvar = boot.laten.
                   offsetvar,
                b, beta, model, link, emmax, eps)

    b_boot[i,]<-bootfit$b
    beta_boot[i,]<-bootfit$latencyfit
    if(bootfit$tau < eps){i<-i+1}
    # i<-i+1
  }

}else{
  # inci off + 0 (laten off = null)

  # sampling according to censored and uncensored
     data
  tempdata<-cbind(Time, Status, X, Z, incidence.
     offsetvar)
  data1<-subset(tempdata, Status==1) #uncensored
```

```
data0<-subset(tempdata, Status==0) #censored

ntemp<-nrow(tempdata)
n1<-nrow(data1)
n0<-nrow(data0)

#samsize<-80 ####flexible!

i<-1
while(i <= nboot){

  #id<-sample(1:ntemp, size = ntemp, replace =
     TRUE)
  id1<-sample(1:n1, size = n1, replace = TRUE) #
     sampling from uncensored
  id0<-sample(1:n0, size = n0, replace = TRUE) #
     sampling from censored

  bootdata<-rbind(data1[id1,], data0[id0,])
  bootZ<-bootdata[,bnm]
  nbdata<-length(bootdata[,1])

  if(model == "ph"){
    bootX<-as.matrix( cbind( rep(1,nbdata) ,
       bootdata[,betanm]) )
  }
  # offset
```

```
      boot.inci.offsetvar<-bootdata[,incidence.nm]

      bootfit<-em(bootdata[,1], bootdata[,2], bootX,
         bootZ,
                  incidence.offsetvar = boot.inci.
                     offsetvar,
                  latency.offsetvar = NULL,
                  b, beta, model, link, emmax, eps)

      b_boot[i,]<-bootfit$b
      beta_boot[i,]<-bootfit$latencyfit
      if(bootfit$tau < eps){i<-i+1}
      # i<-i+1
    }
  }
}else if (!is.null(latency.offsetvar)){
  # 0 + laten off (incidence = NULL)

  # sampling according to censored and uncensored data
  tempdata<-cbind(Time, Status, X, Z, latency.
     offsetvar)
  data1<-subset(tempdata, Status==1) #uncensored
  data0<-subset(tempdata, Status==0) #censored

  ntemp<-nrow(tempdata)
  n1<-nrow(data1)
  n0<-nrow(data0)
```

```
#samsize<-80 ####flexible!

i<-1
while(i <= nboot){

  #id<-sample(1:ntemp, size = ntemp, replace = TRUE)
  id1<-sample(1:n1, size = n1, replace = TRUE) #
     sampling from uncensored
  id0<-sample(1:n0, size = n0, replace = TRUE) #
     sampling from censored

  bootdata<-rbind(data1[id1,], data0[id0,])
  bootZ<-bootdata[,bnm]
  nbdata<-length(bootdata[,1])

  if(model == "ph"){
    bootX<-as.matrix( cbind( rep(1,nbdata) ,
       bootdata[,betanm]) )
  }
  # offset
  boot.laten.offsetvar<-bootdata[,latency.nm]

  bootfit<-em(bootdata[,1], bootdata[,2], bootX,
     bootZ,
              incidence.offsetvar = NULL,
              latency.offsetvar = boot.laten.
```

```
                    offsetvar,
                  b, beta, model, link, emmax, eps)

    b_boot[i,]<-bootfit$b
    beta_boot[i,]<-bootfit$latencyfit
    if(bootfit$tau < eps){i<-i+1}
    # i<-i+1
  }

}else{
  # 0 + 0 (absolutely no offset terms)

  # sampling according to censored and uncensored data
  tempdata<-cbind(Time, Status, X, Z)
  data1<-subset(tempdata, Status==1) #uncensored
  data0<-subset(tempdata, Status==0) #censored

  ntemp<-nrow(tempdata)
  n1<-nrow(data1)
  n0<-nrow(data0)



  i<-1
  while(i <= nboot){

    #id<-sample(1:ntemp, size = ntemp, replace = TRUE)
    id1<-sample(1:n1, size = n1, replace = TRUE) #
```

```
      sampling from uncensored
    id0<-sample(1:n0, size = n0, replace = TRUE) #
      sampling from censored

    bootdata<-rbind(data1[id1,], data0[id0,])
    bootZ<-bootdata[,bnm]
    nbdata<-length(bootdata[,1])

    if(model == "ph"){
      bootX<-as.matrix( cbind( rep(1,nbdata) ,
        bootdata[,betanm]) )
    }

    bootfit<-em(bootdata[,1], bootdata[,2], bootX,
      bootZ,
                incidence.offsetvar = NULL, latency.
                  offsetvar = NULL,
                b, beta, model, link, emmax, eps)

    b_boot[i,]<-bootfit$b
    beta_boot[i,]<-bootfit$latencyfit
    if(bootfit$tau < eps){i<-i+1}
    # i<-i+1
  }
}

b_var<-apply(b_boot, 2, var)
```

```
  beta_var<-apply(beta_boot, 2, var)
  b_std<-sqrt(b_var)
  beta_std<-sqrt(beta_var)
}

########################

class(fit)<-c("sm5")

fit<-list()

fit$logistfit<-logistfit
fit$b<-b
fit$beta<-beta
fit$call<-call
fit$bnm<-bnm
fit$betanm<-betanm
fit$s<-s
fit$uncuredprob<-uncuredprob
fit$Time<-Time
# fit$Spop<-Spop
# fit$surv<-surv.est
fit$iternum<-iternum



if(Var){
  fit$b_var<-b_var
```

```
    fit$b_std<-b_std
    fit$beta_var<-beta_var
    fit$beta_std<-beta_std

    #p-value
    fit$b_zvalue<-fit$b/b_std
    fit$b_pvalue<-(1-pnorm(abs(fit$b_zvalue)))*2

    fit$beta_zvalue<-fit$beta/beta_std
    fit$beta_pvalue<-(1-pnorm(abs(fit$beta_zvalue)))*2
    #z.value and p.value!
  }


  fit
}

# predict
predictsm5<-function(object, newX, newZ,
                     newincioffset = NULL, newlatenoffset
                       = NULL,
                     model = c("ph", "aft"), ...){

  call <- match.call()
  # if(!inherits(object, "sm5")) stop("Object must be
     results of sm5")
```

```
if(is.vector(newZ)){
  newZ=as.matrix(newZ)
}else{
  newZ=cbind(1,newZ)
}

if(is.vector(newX)){
  newX=as.matrix(newX)
}
# only allow one variable to be offset
if(!is.null(newincioffset)){
  incioff.new<-as.matrix(newincioffset)
}
if(!is.null(newlatenoffset)){
  latenoff.new<-as.matrix(newlatenoffset)
}

s0=as.matrix(object$s,ncol=1)
n=nrow(s0)
if(!is.null(newincioffset)){
  #b_matrix<-matrix(object$b,ncol = length(object$b))
  uncuredprob<-exp( object$b%*%t(newZ) + 1%*%t(incioff.
     new) )/
    (1+exp( object$b%*%t(newZ) + 1%*%t(incioff.new) ))
}else{
  uncureprob=exp(object$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(!is.null(newlatenoffset)){

  if(model=='ph')
  {ebetaX=exp(object$beta%*%t(newX) + 1%*%t(latenoff.new
     ) )
  for( i in 1:nrow(newZ))
  {scure[,i]=s0^ebetaX[i]}
  for (i in 1:n){
    for (j in 1:nrow(newX)){
      spop[i,j]=uncureprob[j]*scure[i,j]+(1-uncureprob[j
         ])
    }
  }
  prd=cbind(spop,Time=object$Time)
  }

}else{

  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=='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="predictsm5")
}

# plot
plotsm5<-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

R code for first simulation study design with Weibull 1 data set

```
# test different Weibull distribution baseline simulated
   data

library("survival")

N<-200
simu<-500

#generate data

gen.pi1<-function(px1,px2,b0,b1,b2,b3,beta1,beta2,beta3,w1
   =NULL,w2=NULL,upbound){

  #type <- match.arg(type)

  x1<-rbinom(N,1,px1)
  x2<-rnorm(N,0,1)
  x3<-rbinom(N,1,px2)
```

```
#cure indicator
logit<-cbind(1,x1,x2,x3)%*%c(b0,b1,b2,b3)
pi<-exp(logit)/(1+exp(logit)) #proportion of the uncured
#pi2<-1-exp(logit)/(1+exp(logit))
y<-rbinom(N,1,prob = pi)
#y2<-rbinom(N,1,prob = pi2)

#time to event
u<-runif(N,0,1)
bX<-cbind(x1,x2,x3)%*%c(beta1,beta2,beta3)
#case-1
S0<- exp(log(u)/exp(bX))
#case-2
#S0<-exp(exp(log(log(u))-bX)) #generating NA
#case-3
#S0<- exp(-exp(log(-log(u))-bX))

t<-qweibull(1-S0,shape = w1, scale = w2)

#time to censoring
C<-runif(N,2,upbound)

delta<-as.numeric(t<=C)
Status<-ifelse(y==0,0,delta)
Time<-Status*t+(1-Status)*C
```

```
  mcdata<-data.frame(Status,Time,y,x1,x2,x3)
}

for (i in 1:simu){
  setwd("C:/~/small_wei")

  mcdata1<-gen.pi1(px1 = 0.5, px2 = 0.2, b0=2, b1=-1, b2
     =2, b3=1, beta1=-2, beta2=2, beta3=1, w1=5, w2=5,
     upbound = 50)
  name1 <- paste ("w1data", i,".csv ", sep ="")
  write.csv(mcdata1, file = name1)
}

# cure rate and censoring rate
curewei1<-matrix(rep(0,simu),nrow = simu)
cenwei1<-matrix(rep(0,simu),nrow = simu)

for(i in 1:simu){
  setwd("C:/~/small_wei")
  name1 <- paste ("w1data", i,".csv ", sep ="")
  mcdata1<-read.csv(file = name1)

  #weibull1
  curewei1[i,]<-1-sum(mcdata1$y)/N
  cenwei1[i,]<-1-sum(mcdata1$Status)/N
  }
curewei1_vec<-apply(curewei1, 2,mean)
```

```
cenwei1_vec<-apply(cenwei1, 2, mean)

# the modified EM algorithm
for(i in 1:simu){
  setwd("C:/~/small_wei")
  name1 <- paste ("w1data", i,".csv ", sep ="")

  mcdata1<-read.csv(file = name1)

  fit1<-sm5(formula = Surv(Time,Status)~x1+x2,cureform = ~
     x1+x2, incidence.offset= ~x3, latency.offset= ~x3,
     data = mcdata1, model = "ph",link = "logit", Var =
     TRUE, nboot = 100)

  #b0=2, b1=-1, b2=2, b3=1, beta1=-2, beta2=2, beta3=1
  est1[i,]<-c(fit1$b,fit1$beta)
  estbias1[i,]<-est1[i,]-c(2,-1,2,-2,2)
  var1[i,]<-c(fit1$b_var,fit1$beta_var)
  iternum1[i,]<-fit1$iternum
}

estvector1<-apply(est1, 2, mean)
biasvector1<-apply(estbias1,2,mean)
varvector1<-apply(var1,2,mean)
empirical1<-apply(est1, 2, var)
iter1<-apply(iternum1,2,mean)
```

Appendix C

R code for BMT data analysis

```
# [Real data analysis] Bone marrow transplantation data
#####################################################
library("survival")
library("geecure")
library("intsurv")
library("smcure")
library("survminer")

data(bmt, package = "geecure")

test.data<-na.omit(data.frame(T1 = bmt$T1, d1 = bmt$d1, T2
    = bmt$T2, d2 = bmt$d3,Ta = bmt$TA, da = bmt$da, paage
   = bmt$Z1, doage = bmt$Z2, pasex = bmt$Z3, dosex =
   bmt$Z4, pa.CMV = bmt$Z5, do.CMV = bmt$Z6, wait = bmt$Z7
   , FAB = bmt$Z8, MTX = bmt$Z10))

# smcure
fit.sm<-smcure(formula = Surv(T1,d1) ~ paage + doage +
   pasex + dosex +
                  pa.CMV + do.CMV + wait + FAB + MTX,
```

```
                cureform = ~ paage + doage + pasex + dosex
                   +
                  pa.CMV + do.CMV + wait + FAB + MTX,
                link = "logit", data = test.data, model = "
                   ph", Var = TRUE, nboot = 100)

b<-as.numeric(fit.sm$b)
b_std<-as.numeric(fit.sm$b_sd)

  #p-value test for 1
  b_zvalue<-(b-rep(1,times=length(b)))/b_std
  b_pvalue<-(1-pnorm(abs(b_zvalue)))*2

beta<-as.numeric(fit.sm$beta)
beta_std<-as.numeric(fit.sm$beta_sd)

  beta_zvalue<-(beta-rep(1,times=length(beta)))/beta_std
  beta_pvalue<-(1-pnorm(abs(beta_zvalue)))*2
  #z.value and p.value!



# sm5
fit.sm5<-sm5(formula = Surv(T1,d1) ~ paage + doage + pasex
    + dosex +
                pa.CMV + do.CMV + wait + FAB + MTX,
              cureform = ~ paage + doage + pasex + dosex +
```

```
                pa.CMV + do.CMV + wait + FAB + MTX,
              link = "logit", data = test.data, model = "ph
                 ", Var = TRUE, nboot = 100)

# discuss an offset

fit.sm5offset<-sm5(formula = Surv(T1,d1) ~ MTX ,
                   cureform = ~ pa.CMV,
                   incidence.offset = NULL, latency.offset
                       = ~ FAB,
                   link = "logit", data = test.data, model
                       = "ph", Var = TRUE, nboot = 100)
fit.smoff<-smcure(formula = Surv(T1,d1) ~ FAB + MTX,
                  cureform = ~ pa.CMV ,
                  link = "logit", data = test.data, model
                     = "ph", Var = TRUE, nboot = 100)

# graph

# km curve
## overall
overall<-survfit( Surv(T1,d1)~1, data = test.data)
png("C:/~/bmt_overall_curve.png",
    width = 400, height = 300)
ggsurvplot(overall, palette= 'black', legend = "none",
   size = 0.75,
           xlab = "Time (in days)")
```

```
dev.off()

## based on MTX
km.mtx<-survfit( Surv(T1,d1)~ MTX, data = test.data)
png("C:/~/bmt_mtx_curve.png",
    width = 400, height = 300)
ggsurvplot(km.mtx, palette= c("black", "red"), size =
   0.75,
           xlab = "Time (in days)")
dev.off()

## predicted sm5 mtx
png("C:/~/bmt_mtx_sm5predict.png",
    width = 400, height = 300)

pred.sm5<-predictsm5(fit.sm5offset, newX = c(0,1), newZ =
   c(0,1), data = test.data, model = "ph")
plotsm5(pred.sm5, xlab = "Time (in days)", ylab = "
   Survival probability")
legend("topright", legend = c(0,1), lty=1:(ncol(pred)-1),
   bty="n")

dev.off()
```