PROGRAMS FOR SEMIPARAMETRIC COX REGRESSION WITH CUBIC M-SPLINE

TAKESHI EMURA*
Graduate Institute of Statistics, National Central University, Taiwan
email: takeshiemura@gmail.com

JIA-HAN SHIH
Graduate Institute of Statistics, National Central University, Taiwan
email: tommy355097@gmail.com

Hazard models with cubic spline functions have a number of advantages to the standard Cox model. However, existing software packages for fitting the cubic spline models are not simple enough to analyze right-censored data. In this paper, we introduce methodologies and R programs to fit the cubic spline functions using penalized likelihood inference. For illustration, we analyze a life test dataset on electrical components and a gene expression dataset on lung cancer patients.

Keywords: Right-censoring, Cox regression, Life test, Smoothing.

1. Introduction

In survival analysis, a semiparametric model is often defined by a hazard function that imposes a specific form of covariate effects without specifying a distributional form. The most popular model is the Cox proportional hazards model (Cox 1972) [2]. Semiparametric models are more flexible and often fit better to data than parametric models by allowing the data to determine its functional form. However, the standard Cox regression method can be ineffective for data with small samples or heavy censoring. In addition, the Cox partial likelihood method (Cox 1972) [2] does not directly produce the estimator of the baseline hazard function.

A semiparametric hazard model with cubic spline functions imposes some mild assumptions on the hazard function without restricting too much the shape of the hazard function. The use of the splines was proposed by O'Sullivan (1998) [8]. The spline model can handle small samples and heavy censoring by reducing the number of knots and adopting a penalized likelihood method. The

[*] Corresponding Author

book of Commenges and Jacqmin-Gadda (2015) [1] reviews the spline-based method, and its applications to complex survival models.

Some software packages for fitting the cubic spline models were developed, such as *PHMPL* (Joly et al. 1999) [3] and *frailtypack* (Rondeau and Gonzalez 2005) [10]. Unfortunately, these packages are not tailored to right-censored data. *PHMPL* is designed for interval censored and left-truncated data while *frailtypack* is designed for frailty models. Consequently, they are not simple enough to analyze right-censored data, including Type I censored data that frequently appear in reliability. In this paper, we introduce methodologies and R programs to fit the cubic spline functions using penalized likelihood inference. We analyze life test data and lung cancer data for illustrations. All R functions introduced in this paper, *M.spline( )*, *I.spline( )*, and *splineCox.reg( )*, were made available in the *joint.Cox* R package (Emura 2019) [7].

2. Cubic M-spline model

This section reviews the cubic spline models for a baseline hazard function and a penalized likelihood approach for fitting right-censored data.

2.1. *Data structure*

This section summarizes the basic notations. For each unit *i*, we consider *random variables*, defined as

- X_i: event time
- C_i: censoring time

Due to *censoring*, either one of X_i or C_i is observed. Thus, what we observe are $T_i = \min\{ X_i, C_i \}$ and $\delta_i = \mathbf{I}(X_i \le C_i)$, where $\mathbf{I}(\cdot)$ is the indicator function. Survival data often include *covariates*, such as the stress level. With covariates, survival data consist of { $(T_i, \delta_i, \mathbf{Z}_i)$, $i = 1, ..., n$ }, where

- $\mathbf{Z}_i = (Z_{i1}, ..., Z_{ip})'$: *p*-dimensional covariates.

Throughout, we impose the independent censoring assumption: X_i and C_i are conditionally independent given $\mathbf{Z}_i$.

2.2. *The spline model*

Let $S_X(t \,|\, \mathbf{Z}_i) = \Pr(X_i > t \,|\, \mathbf{Z}_i)$ be the conditional survival function given $\mathbf{Z}_i$. Also, let $\lambda_X(t \,|\, \mathbf{Z}_i) = -d \log S_X(t \,|\, \mathbf{Z}_i) / dt$ be the conditional hazard function. We impose the Cox proportional hazards model (Cox 1972) [2]

$$\lambda(t \,|\, \mathbf{Z}_i) = \lambda_0(t; \mathbf{h}) \exp(\boldsymbol{\beta}' \mathbf{Z}_i),$$

where the baseline hazard function is specified as

$$\lambda_0(t;\mathbf{h}) = \sum_{\ell=1}^{L} h_\ell M_\ell(t),$$

where h_ℓ 's are positive parameters and $M_\ell(t)$'s are called the M-spline basis functions (Ramsay 1988) [9]. The number of bases L is pre-specified.

One has the baseline cumulative hazard function and survival function

$$\Lambda_0(t;\mathbf{h}) = \sum_{\ell=1}^{L} h_\ell I_\ell(t), \qquad S_0(t;\mathbf{h}) = \exp\left[-\sum_{\ell=1}^{L} h_\ell I_\ell(t)\right],$$

where $I_\ell(t)$'s are integrations of $M_\ell(t)$'s, called the I-spline basis functions (Ramsay 1988) [9].

Following Emura et al. (2017) [6], we suggest choosing the number $L=5$. With this number, the baseline hazard function becomes more flexible than those produced by the two-parameter models (e.g. Weibull and lognormal models).

The actual formulas of $M_\ell(t)$'s are

$$M_1(t) = -\frac{4\mathbf{I}(\xi_1 \le t < \xi_2)}{\Delta} z_2(t)^3, \qquad M_5(t) = \frac{4\mathbf{I}(\xi_2 \le t < \xi_3)}{\Delta} z_2(t)^3,$$

$$M_2(t) = \frac{\mathbf{I}(\xi_1 \le t < \xi_2)}{2\Delta}\{\, 7z_1(t)^3 - 18z_1(t)^2 + 12z_1(t) \,\} - \frac{\mathbf{I}(\xi_2 \le t < \xi_3)}{2\Delta} z_3(t)^3,$$

$$M_3(t) = \frac{\mathbf{I}(\xi_1 \le t < \xi_2)}{\Delta}\{\, -2z_1(t)^3 + 3z_1(t)^2 \,\} + \frac{\mathbf{I}(\xi_2 \le t < \xi_3)}{\Delta}\{\, 2z_2(t)^3 - 3z_2(t)^2 + 1 \,\},$$

$$M_4(t) = \frac{\mathbf{I}(\xi_1 \le t < \xi_2)}{2\Delta} z_1(t)^3 + \frac{\mathbf{I}(\xi_2 \le t < \xi_3)}{2\Delta}\{\, -7z_2(t)^3 + 3z_2(t)^2 + 3z_2(t) + 1 \,\}\cdot$$

The actual formulas of $I_\ell(t)$'s are

$$I_1(t) = 1 - z_2(t)^4 \mathbf{I}(\xi_1 \le t < \xi_2), \qquad I_5(t) = z_2(t)^4 \mathbf{I}(\xi_2 \le t < \xi_3),$$

$$I_2(t) = \{\, \frac{7}{8} z_1(t)^4 - 3z_1(t)^3 + 3z_1(t)^2 \,\}\mathbf{I}(\xi_1 \le t < \xi_2) + \{\, 1 - \frac{1}{8} z_3(t)^4 \,\}\mathbf{I}(\xi_2 \le t < \xi_3),$$

$$I_3(t) = \{\, -\frac{1}{2} z_1(t)^4 + z_1(t)^3 \,\}\mathbf{I}(\xi_1 \le t < \xi_2) + \{\, \frac{1}{2} + \frac{1}{2} z_2(t)^4 - z_2(t)^3 + z_2(t) \,\}\mathbf{I}(\xi_2 \le t < \xi_3),$$

$$I_4(t) = \frac{1}{8} z_1(t)^4 \mathbf{I}(\xi_1 \le t < \xi_2) + \{\, \frac{1}{8} - \frac{7}{8} z_2(t)^4 + \frac{1}{2} z_2(t)^3 + \frac{3}{4} z_2(t)^2 + \frac{1}{2} z_2(t) \,\}\mathbf{I}(\xi_2 \le t < \xi_3),$$

where $\xi_1 < \xi_2 < \xi_3$, $\Delta = \xi_2 - \xi_1 = \xi_3 - \xi_2$, and $z_i(t) = (t - \xi_i)/\Delta$, $i = 1, 2, 3$.

The value ξ_1 is the lower limit for *t*, ξ_3 is the upper limit for *t*, and $\xi_2 = (\xi_1 + \xi_3)/2$ is the intermediate for *t*. In data analysis, one can choose $\xi_1 = \min(T_j)$ and $\xi_3 = \max(T_j)$. Figure 1 displays the M- and I-spline basis functions with $L=5$ and the knots $\xi_1 = 1$, $\xi_2 = 2$, and $\xi_3 = 3$. The *joint.Cox* package (Emura 2019) provides a function *M.spline( )* for $M_\ell(t)$ and *I.spline( )* for $I_\ell(t)$ for $L=5$. See Appendix A of Emura and Chen (2018) for details.

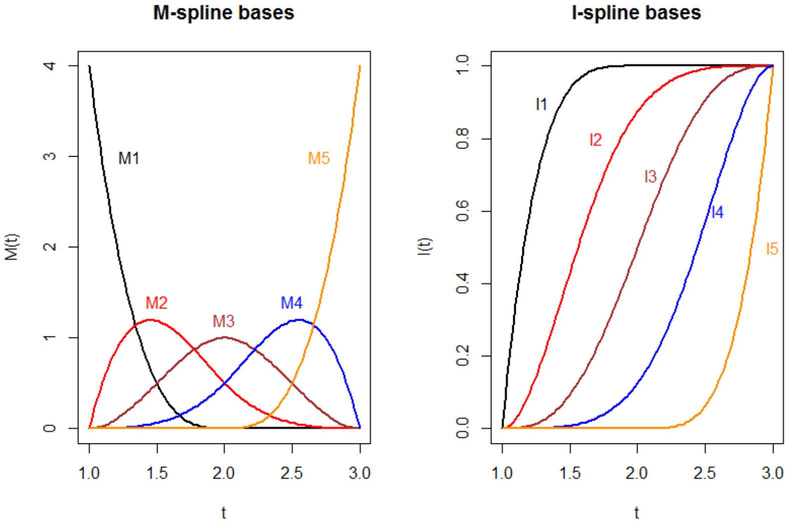

Figure 1: M-spline basis functions (left-panel) and I-spline basis functions (right-panel) with the number $L=5$ and knots $\xi_1=1$, $\xi_2=2$, and $\xi_3=3$.

For instance, if one wishes to compute $M_\ell(t)$'s at $t=1.5$ and $t=2.5$ under the knots $\xi_1=1$ and $\xi_3=3$, enter the commands:

```
M.spline(c(1.5, 2.5),xi1=1,xi3=3)
```

Below is output.

```
      M1     M2        M3     M4        M5
[1,]  0.5    1.1875   0.5    0.0625   0.0
[2,]  0.0    0.0625   0.5    1.1875   0.5
```

3. Penalized likelihood

Based on the observed data $\{\,(\,T_i,\,\delta_i,\,\mathbf{Z}_i\,),\ i=1,\ldots,n\,\}$, the log-likelihood function is written as

$$\ell(\boldsymbol{\varphi})=\sum_{i=1}^{n}[\,\delta_i\{\,\log\lambda_0(T_i;\mathbf{h})+\boldsymbol{\beta}'\mathbf{Z}_i\,\}-\Lambda_0(T_i;\mathbf{h})\exp(\boldsymbol{\beta}'\mathbf{Z}_i)\,],$$

where $\boldsymbol{\varphi}=(\mathbf{h},\boldsymbol{\beta})$ are the parameters to be maximized. When the spline function is applied, it is customary to apply a penalized likelihood approach. We define the complexity of a function f is through the *roughness* defined as $\int\ddot{f}(t)^2dt$, where $\ddot{f}(t)=d^2f(t)/dt^2$. We then maximize the *penalized likelihood*

$$\ell_{PL}(\boldsymbol{\varphi}) = \ell(\boldsymbol{\varphi}) - \kappa \int \ddot{\lambda}_0(t;\mathbf{h})^2 dt\,. \tag{1}$$

where $\kappa > 0$ is a given value, called a smoothing parameter. Let $\hat{\boldsymbol{\varphi}}$ be the maximizer of Equation (1) for a given value of κ.

Under the five-parameter splines, it can be shown (Emura et al. 2017) [7] that

$$\int \ddot{\lambda}_0(t;\mathbf{h})^2 dt = \mathbf{h}'\Omega\mathbf{h}, \qquad \Omega = \begin{bmatrix} 192 & -132 & 24 & 12 & 0 \\ -132 & 96 & -24 & -12 & 12 \\ 24 & -24 & 24 & -24 & 24 \\ 12 & -12 & -24 & 96 & -132 \\ 0 & 12 & 24 & -132 & 192 \end{bmatrix}.$$

Hence, the penalized log-likelihood is simplified as

$$\ell_{PL}(\boldsymbol{\varphi}) = \ell(\boldsymbol{\varphi}) - \kappa\mathbf{h}'\Omega\mathbf{h}$$

The *joint.Cox* R package (Emura 2019) [7] provides a function *splineCox.reg( )* to compute $\hat{\boldsymbol{\varphi}}$. The function automatically selects the value of κ by maximizing the likelihood cross-validation (LCV) criterion

$$LCV = \hat{\ell} - \mathrm{tr}\{\,\hat{H}_{PL}^{-1}\hat{H}\,\},$$

where $\hat{\ell} = \ell(\hat{\boldsymbol{\varphi}})$, $\hat{H}_{PL}$ is the Hessian matrix of $\ell_{PL}(\boldsymbol{\varphi})$ at $\hat{\boldsymbol{\varphi}}$, and $\hat{H}$ is the Hessian matrix of $\ell(\boldsymbol{\varphi})$ at $\hat{\boldsymbol{\varphi}}$. That is, $\hat{H}_{PL} = H_{PL}(\hat{\boldsymbol{\varphi}})$, where $H_{PL}(\boldsymbol{\varphi}) = \partial^2\ell_{PL}(\boldsymbol{\varphi})/\partial\boldsymbol{\varphi}^2$ and $\hat{H} = H(\hat{\boldsymbol{\varphi}})$, where $H(\boldsymbol{\varphi}) = \partial^2\ell(\boldsymbol{\varphi})/\partial\boldsymbol{\varphi}^2$. The term $\mathrm{tr}\{\,\hat{H}_{PL}^{-1}\hat{H}\,\}$ is the effective degrees of freedom, a decreasing function in κ. The two Hessian matrices are related through

$$\hat{H} = \hat{H}_{PL} + 2\kappa \begin{bmatrix} O_{p\times p} & O_{p\times 5} \\ O_{5\times p} & \Omega \end{bmatrix},$$

where O is a zero matrix and p is the dimension of covariates.

The LCV was suggested by O'Sullivan (1998) [8]. It is a criterion capable of choosing the best values of κ as well as selecting the best subset of covariates. Hence, the LCV plays a similar role as the AIC for model selection.

The *splineCox.reg( )* function provides the plots of LCV and the optimized values for κ by a grid search (the grid must be given by user). When looking at the plot, following properties must be checked: (i) $\hat{\ell}$ is smoothly decreasing in κ, (ii) the degrees of freedom $\mathrm{tr}\{\,\hat{H}_{PL}^{-1}\hat{H}\,\}$ decreases from $p+5$ to $p+2$. If (i) and (ii) are not met, the grid is wrong.

Finally, the penalized likelihood (PL) estimator is defined as

$$\hat{\boldsymbol{\varphi}} = (\hat{\boldsymbol{\beta}}^{\mathrm{PL}}, \hat{\mathbf{h}}^{\mathrm{PL}}) = \arg\max\left\{\ell(\boldsymbol{\beta},\mathbf{h}) - \hat{\kappa}\mathbf{h}'\Omega\mathbf{h}\right\},$$

where $\hat{\kappa}$ is the optimized value.

Interval estimates for $\boldsymbol{\varphi}$ follow from the asymptotic theory of maximum likelihood estimators. The *information matrix* is defined as $i(\hat{\boldsymbol{\varphi}})=-H_{PL}(\hat{\boldsymbol{\varphi}})$. For the *j*-th component $\hat{\varphi}_j$ of $\hat{\boldsymbol{\varphi}}$, the standard error (SE) is defined as $SE(\hat{\varphi}_j)=\sqrt{\{\,i^{-1}(\hat{\boldsymbol{\varphi}})\,\}_{jj}}$, where $\{\,i^{-1}(\hat{\boldsymbol{\varphi}})\,\}_{jj}$ is the *j*-th diagonal element of the inverse information matrix. Then, the 95% confidence interval (CI) is obtained as $\hat{\varphi}_j \pm 1.96\times SE(\hat{\varphi}_j)$.

For our interest, the estimate of the baseline hazard function is $\hat{\lambda}_0(t;\hat{\mathbf{h}}^{PL})=\mathbf{M}'(t)\hat{\mathbf{h}}^{PL}$, where $\mathbf{M}'(t)=[M_1(t),\ldots,M_L(t)]$. By applying the Delta method, we obtain the SE of the baseline hazard function as $SE\{\hat{\lambda}_0(t;\hat{\mathbf{h}}^{PL})\}=\sqrt{-\mathbf{M}'(t)\{H_{PL}^{-1}(\hat{\boldsymbol{\varphi}})\}_{\mathbf{h}}\mathbf{M}(t)}$. Then, the 95% CI of the baseline hazard function is $\hat{\lambda}_0(t;\hat{\mathbf{h}}^{PL})\pm 1.96\times SE\{\hat{\lambda}_0(t;\hat{\mathbf{h}}^{PL})\}$.

4. Data examples

We shall analyze two datasets for illustration.

4.1. *Life test data on electrical components*

We consider the life test data of Zhou et al. (2018) [11]. The data contain failure times of *n*=64 electrical components obtained under three temperature levels (131.5 °C, 142.5 °C, and 158 °C). We shall consider an ordinal covariate (*Z*=2 for 131.5 °C; *Z*=1 for 142.5 °C; *Z*=0 for 158 °C). Nine samples were right-censored at the pre-specified period of 1540 hours (Type I censoring). The data are plotted in Figure 2 (left panel).

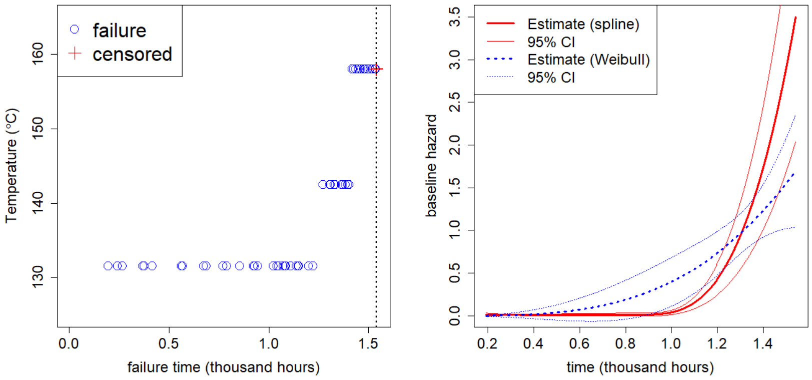

Figure 2. Left panel: The life test data on electrical components (Zhou et al. 2018). Right panel: The estimated baseline hazard function with 95% CI.

We use the three methods to analyze the data: (a) the usual Cox regression method; (b) the Weibull regression method; (c) the penalized likelihood method with spline. We use the following R codes to perform our analysis.

```
library(survival); library(joint.Cox); library(numDeriv)

t1=c(1.043, 0.376, 0.417, 0.675, 0.924, 0.930, 1.137, 0.194, 1.050, 0.268,
     1.221, 0.415, 1.082, 0.368, 0.563, 0.856, 1.152, 0.241, 1.087, 1.149, 0.791,
     1.152, 0.945, 0.770, 1.085, 1.202, 0.569, 0.686, 1.023, 1.072, 0.561, 1.110)
t2=c(1.311, 1.404, 1.271, 1.367, 1.305, 1.396, 1.329, 1.335, 1.384, 1.371)
t3=c(1.419, 1.438, 1.475, 1.538, 1.425, 1.520, 1.461, 1.486, 1.497, 1.510, 1.451,
     1.482, 1.534, 1.540, 1.540, 1.540, 1.540, 1.540, 1.540, 1.540, 1.540, 1.540)
n1=length(t1);n2=length(t2);n3=length(t3)
t.event=c(t1,t2,t3) ## failure time
event=c(rep(1,n1),rep(1,n2),rep(1,n3-9),rep(0,9)) ## 1=failure; 0=censor
Z=c(rep(2,n1),rep(1,n2),rep(0,n3)) ## three temperature levels
xi1=min(t.event); xi3=max(t.event)
tvec=seq(xi1,xi3,length.out=500)

coxph(Surv(t.event,event)~Z)
res=splineCox.reg(t.event,event,Z,kappa=seq(0,50,length.out=10)); res

### data ###
par(mfrow=c(1,1),cex.lab = 1.5,cex.axis = 1.5,mar = c(5,5,3,2))
xnames="failure time (thousand hours)"
ynames=expression(paste("Temperature (",degree,"C)"))
xlims=c(0,1.55); ylims=c(125,165)
plot(t1,rep(131.5,n1),ylab=ynames,xlab=xnames,xlim=xlims,ylim=ylims,cex=2,col="blue")
points(t2,rep(142.5,n2),cex=2,col="blue")
points(t3[1:(n3-9)],rep(158,n3-9),cex=2,col="blue")
points(rep(1.54,9),rep(158,9),cex=2,col="red",pch=3,lwd=2)
abline(v=1.54,lty=3,lwd=2)
legend("topleft",c("failure","censored"),col=c("blue","red"),cex=2,pch=c(1,3))

### hazard (spline) ###
splhazard=function(time){as.numeric(M.spline(time,xi1,xi3)%*%(res$h))}
splLow=function(time){
  r_V=M.spline(time,xi1,xi3)%*%(res$h_var)%*%t(M.spline(time,xi1,xi3))
  as.numeric(M.spline(time,xi1,xi3)%*%(res$h)-1.96*sqrt(diag(r_V)))
}
splUp=function(time){
  r_V=M.spline(time,xi1,xi3)%*%(res$h_var)%*%t(M.spline(time,xi1,xi3))
  as.numeric(M.spline(time,xi1,xi3)%*%(res$h)+1.96*sqrt(diag(r_V)))
}

### estimated hazard (spline) ###
plot(tvec,sapply(tvec,splhazard),lwd=3,xlab="time (thousand hours)",
     ylab="baseline hazard",xlim=c(xi1,xi3),type="l",col="red")
lines(tvec,sapply(tvec,splLow),col="red")
lines(tvec,sapply(tvec,splUp),col="red")

### weibull regression ###
fit=survreg(Surv(t.event,event)~Z,dist="weibull"); summary(fit)
```

```
alpha1=as.numeric(fit$coefficients[1])
alpha2=as.numeric(fit$coefficients[2])
sigma=fit$scale

covhat=diag(c(1,1,sigma),3,3)%*%fit$var%*%diag(c(1,1,sigma),3,3)
beta2=-alpha2/sigma; beta2
trans=c(0,-1/sigma,alpha2/sigma^2)
as.numeric(beta2/sqrt(t(trans)%*%covhat%*%trans))

weihazard=function(time,para) {
  shape=1/para[1]; scale=exp(para[2])
  dweibull(time,shape,scale)/(1-pweibull(time,shape,scale))
}
weiLow=function(time,para){
  trans=grad(weihazard,para,time=time)
  V=as.numeric(t(trans)%*%covhat[-2,-2]%*%trans)
  weihazard(time,para=para)-1.96*sqrt(V)
}
weiUp=function(time,para){
  trans=grad(weihazard,para,time=time)
  V=as.numeric(t(trans)%*%covhat[-2,-2]%*%trans)
  weihazard(time,para=para)+1.96*sqrt(V)
}

lines(tvec,sapply(tvec,weihazard,para=c(sigma,alpha1)),lwd=3,col="blue",lty=3)
lines(tvec,sapply(tvec,weiLow,para=c(sigma,alpha1)),lty=3,col="blue")
lines(tvec,sapply(tvec,weiUp,para=c(sigma,alpha1)),lty=3,col="blue")
lnames=c("Estimate (spline)","95% CI","Estimate (Weibull)","95% CI")
lcol=c("red","red","blue","blue")
legend("topleft",lnames,lwd=c(3,1,3,1),lty=c(1,1,3,3),col=lcol,cex=1.5,bg="white")
```

The outputs are shown below. The usual Cox regression method does not converge, though the estimate is still obtained as $\hat{\boldsymbol{\beta}} = 21.34$ (Z-value=0.005). The Weibull regression gives $\hat{\boldsymbol{\beta}} =$ 1.268 (Z-value=6.651). The penalized likelihood method with spline produces $\hat{\boldsymbol{\beta}}^{PL}$ = 2.092 (Z-value=13.4) under $\kappa = 0$. Hence, the Weibull regression and penalized likelihood methods give more meaningful results including the confidence intervals than the Cox regression method. The estimated baseline hazard functions are plotted in Figure 2 (right panel). Both the spline and Weibull models yield the increasing hazard rate over time, showing aging effects for a long time of use. However, the spline model exhibits a higher risk than the Weibull model did after 1300 hours.

```
> coxph(Surv(t.event,event)~Z)
Call: coxph(formula = Surv(t.event, event) ~ Z)
      coef          exp(coef)    se(coef)       z         p
Z    2.134e+01   1.859e+09    4.207e+03    0.005    0.996

Likelihood ratio test=120.1  on 1 df, p=< 2.2e-16
n= 64, number of events= 55
```

```
Warning message:
In fitter(X, Y, strats, offset, init, control, weights = weights,  :
  Loglik converged before variable  1 ; beta may be infinite.

> beta2=-alpha2/sigma; beta2
[1] 1.267821

> splineCox.reg(t.event,event,Z,kappa=seq(0,50, length = 10))
$beta
estimate        SE                Lower         Upper
2.0921975     0.1561384    1.7861662    2.3982289

$h
[1] 2.557855e-03 1.700781e-03 9.770775e-03 3.984792e-12 5.884791e-01

$h_var
             [,1]          [,2]          [,3]          [,4]          [,5]
[1,]  2.715492e-06 -1.382187e-06 -1.426308e-07 -4.314664e-09  4.084840e-05
[2,] -1.382187e-06  1.381472e-05 -8.955892e-07  8.477369e-09  2.423700e-04
[3,] -1.426308e-07 -8.955892e-07  1.329267e-07 -2.629109e-08 -3.669841e-05
[4,] -4.314664e-09  8.477369e-09 -2.629109e-08 -1.323472e-11 -2.490242e-07
[5,]  4.084840e-05  2.423700e-04 -3.669841e-05 -2.490242e-07  2.005614e-02
$kappa
[1] 0
$DF
[1] 6
$LCV
[1] 6.952789
```

4.2. *Gene expression data from lung cancer patients*

We consider a gene expression data available in the *compound.Cox* R package (Emura et al. 2019) [7]. The data contain time-to-death of *n*=125 lung cancer patients, as well as gene expression levels (*Z*=1, 2, 3, and 4) of *ZNF264*. 87 samples were right-censored either by dropout or the end of study. The data are plotted in Figure 3 (left panel).

We use the three methods to analyze the data: (a) the Cox regression method; (b) the Weibull regression method; (c) the penalized likelihood method with spline. We use the following R codes to perform our analysis.

```
library(survival); library(joint.Cox); library(numDeriv); library(compound.Cox)

data("Lung")
t.event=Lung$t.vec; event=Lung$d.vec; Z=Lung$ZNF264
xi1=min(t.event); xi3=max(t.event)
tvec=seq(xi1,xi3,length.out=500)

coxph(Surv(t.event,event)~Z)
res=splineCox.reg(t.event,event,Z,kappa=seq(10,1e+8,length = 10)); res

### data ###
```

```
par(mfrow=c(1,1),cex.lab = 1.5,cex.axis = 1.5,mar = c(5,5,3,2))
dt=t.event[event==1]; st=t.event[event==0]
dz=Z[event==1]; sz=Z[event==0]
plot(dt,dz,xlab="time-to-death (months)",ylab="gene expressions",
    xlim=c(0,60),ylim=c(0.5,4.5),col="blue",cex=2)
points(st,sz,pch=3,cex=2,col="red")
legend(35,3.5,c("death","censored"),col=c("blue","red"),cex=2,pch=c(1,3))

### hazard (spline) ###
splhazard=function(time){as.numeric(M.spline(time,xi1,xi3)%*%(res$h))}
splLow=function(time){
  r_V=M.spline(time,xi1,xi3)%*%(res$h_var)%*%t(M.spline(time,xi1,xi3))
  as.numeric(M.spline(time,xi1,xi3)%*%(res$h)-1.96*sqrt(diag(r_V)))
}
splUp=function(time){
  r_V=M.spline(time,xi1,xi3)%*%(res$h_var)%*%t(M.spline(time,xi1,xi3))
  as.numeric(M.spline(time,xi1,xi3)%*%(res$h)+1.96*sqrt(diag(r_V)))
}

### estimated hazard (spline) ###
plot(tvec,sapply(tvec,splhazard),lwd=3,xlab="time (months)",
    ylab="baseline hazard",xlim=c(xi1,xi3),ylim=c(0,0.03),type="l",col="red")
lines(tvec,sapply(tvec,splLow),col="red")
lines(tvec,sapply(tvec,splUp),col="red")

### weibull regression ###
fit=survreg(Surv(t.event,event)~Z,dist="weibull"); summary(fit)
alpha1=as.numeric(fit$coefficients[1])
alpha2=as.numeric(fit$coefficients[2])
sigma=fit$scale

covhat=diag(c(1,1,sigma),3,3)%*%fit$var%*%diag(c(1,1,sigma),3,3)
beta2=-alpha2/sigma; beta2
trans=c(0,-1/sigma,alpha2/sigma^2)
as.numeric(beta2/sqrt(t(trans)%*%covhat%*%trans))

weihazard=function(time,para) {
  shape=1/para[1]; scale=exp(para[2])
  dweibull(time,shape,scale)/(1-pweibull(time,shape,scale))
}
weiLow=function(time,para){
  trans=grad(weihazard,para,time=time)
  V=as.numeric(t(trans)%*%covhat[-2,-2]%*%trans)
  weihazard(time,para=para)-1.96*sqrt(V)
}
weiUp=function(time,para){
  trans=grad(weihazard,para,time=time)
  V=as.numeric(t(trans)%*%covhat[-2,-2]%*%trans)
  weihazard(time,para=para)+1.96*sqrt(V)
}

lines(tvec,sapply(tvec,weihazard,para=c(sigma,alpha1)),lwd=3,col="blue",lty=3)
lines(tvec,sapply(tvec,weiLow,para=c(sigma,alpha1)),lty=3,col="blue")
lines(tvec,sapply(tvec,weiUp,para=c(sigma,alpha1)),lty=3,col="blue")
lnames=c("Estimate (Weibull)","95% CI","Estimate (spline)","95% CI")
```

```
lcol=c("blue","blue","red","red")
legend("topleft",lnames,lwd=c(3,1,3,1),lty=c(3,3,1,1),col=lcol,cex=1.5,bg="white")
```

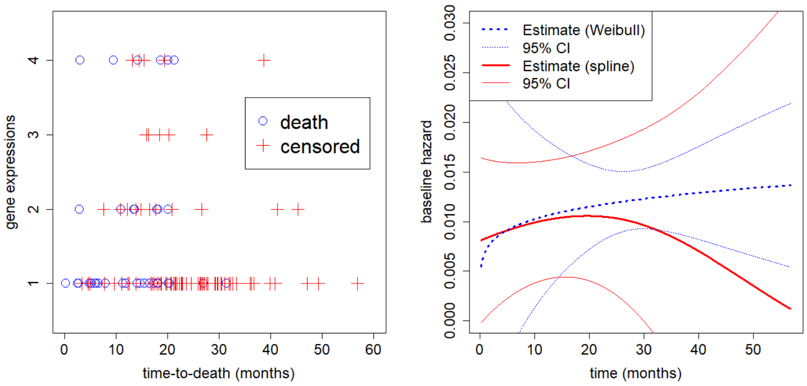

Figure 3. Left panel: The lung cancer data (Emura et al. 2019)
Right panel: The estimated baseline hazard function with 95% CI.

The outputs are shown below. The usual Cox regression method gives $\hat{\boldsymbol{\beta}}=0.2498$ (Z-value=1.638). The Weibull regression method gives $\hat{\boldsymbol{\beta}}=0.2541$ (Z-value=1.664). The penalized likelihood method with spline produces $\hat{\boldsymbol{\beta}}^{PL}$ = 0.2780 (Z-value=1.771) under $\kappa=$ 55555560. Overall, the three methods produce similar results. The Weibull regression and penalized likelihood methods provide the estimated baseline hazard functions (right panel of Figure 3). The Cox regression method, however, does not. Figure 3 shows that the spline model gives a non-monotonic hazard function, having a peak around 20 months. On the other hand, the Weibull model yields a monotonically increasing hazard function. In this example, the Weibull model is not flexible enough to model the non-monotonic hazard function that often arises in survival data from cancer patients.

```
> coxph(Surv(t.event,event)~Z)
Call: coxph(formula = Surv(t.event, event) ~ Z)

    coef       exp(coef)   se(coef)     z          p
Z   0.2498    1.2838        0.1526     1.638     0.101

Likelihood ratio test=2.37  on 1 df, p=0.1235
n= 125, number of events= 38

> beta2=-alpha2/sigma; beta2
[1] 0.2541164

> res=splineCox.reg(t.event,event,Z,kappa=seq(10,1e+8,length = 10)); res
$beta
estimate         SE                    Lower                Upper
```

```
0.27803568    0.15698364      -0.02965225      0.58572361

$h
[1] 0.057477641 0.145246859 0.177059265 0.060750756 0.008535177
$h_var
             [,1]         [,2]         [,3]        [,4]          [,5]
[1,]  8.995618e-04 0.0009568417 0.0001134760 0.000038603 -3.979023e-05
[2,]  9.568417e-04 0.0024124265 0.0006949548 0.001088542  5.241872e-04
[3,]  1.134760e-04 0.0006949548 0.0052267759 0.003372572  1.632282e-03
[4,]  3.860300e-05 0.0010885421 0.0033725722 0.026710185  1.788880e-02
[5,] -3.979023e-05 0.0005241872 0.0016322816 0.017888803  1.292160e-02
$kappa
[1] 55555560
$DF
[1] 3.091727
$LCV
[1] -198.1711
```

5. Concluding remarks

In this paper, our major objective is to estimate the regression parameters and the baseline hazard function simultaneously for right-censored data. However, this cannot be done by the usual Cox regression method. Therefore, we propose a penalized likelihood method with spline to deal with this problem. We analyze the life test data, where the Cox regression does not converge due to small samples, yet the Weibull and spline methods give meaningful results. We also analyze the gene expression data, where all three methods produce similar estimates for a regression coefficient. In this data example, the spline method has the advantage of getting the estimate of the baseline hazard function. Our method is implemented in the *joint.Cox* R package (Emura 2019) [7].

References

[1] D. Commenges, H. Jacqmin-Gadda, Dynamical Biostatistical Models, CRC Press, London, 2015.

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

[3] P. Joly, L. Letenneur, A. Alioum, D. Commenges. PHMPL: a computer program for hazard estimation using a penalized likelihood method with interval-censored and left-truncated data. *Computer Methods and Programs in Biomedicine* 60(3), 225-231, 1999.

[4] T. Emura. joint.Cox: joint frailty-copula models for tumour progression and death in meta-analysis. *CRAN*, 2019.

[5] T. Emura, Y.H. Chen. Analysis of Survival Data with Dependent Censoring, Copula-based Approaches. JSS Research Series in Statistics, Springer, Singapore, 2018

[6] T. Emura, M. Nakatochi, K. Murotani, V. Rondeau. A joint frailty-copula model between tumour progression and death for meta-analysis. *Stat Methods Med Res*, 26 (6) 2649-66, 2017.

[7] T. Emura, S. Matsui, H.Y. Chen. compound.Cox: univariate feature selection and compound covariate for predicting survival. *Comput Methods Programs Biomed*, 168: 21-37, 2019.

[8] F. O'Sullivan. Fast computation of fully automated log-density and log-hazard estimation. *SIAM J Sci Stat Comput*; 9: 363-379, 1998.

[9] J.O. Ramsay. Monotone regression splines in action. *Statistical Science,* 3(4), 425-41, 1988.

[10] V. Rondeau, R. Gonzalez. frailtypack: A computer program for the analysis of correlated failure time data using penalized likelihood estimation. *Comput Methods Programs Biomed* 80 (2); 154-164, 2005.

[11] Y. Zhou, Z. Lu, Y. Shi, K. Cheng. The copula-based method for statistical analysis of step-stress accelerated life test with dependent competing failure modes. *Proc. Inst. Mech. Eng, Part O: Journal of Risk and Reliability*, 1748006X18793251, 2018.