Introduction to SurvPHspline R package

1. Overview

This vignette gives an overview of the `SurvPHspline` R package which fits a semiparametric proportional hazards model (PH), proposed in Withana Gamage *et al.* (2022+), to arbitrarily censored data subject to left-truncation via an EM algorithm. The package provides a function `SurvPHspline.EM()` that uses the following arguments;

```
SurvPHspline.EM(d1, d2, d3, Li, Ri,Ei, Xp, n.int, order, g0, b0, tol,
t.seq, equal = FALSE)
```

- `d1`: vector indicating whether an observation is exactly observed (1) or not (0).
- `d2`: vector indicating whether an observation is interval-censored (1) or not (0).
- `d3`: vector indicating whether an observation is right-censored (1) or not (0).
- `Li`: the left endpoint of the observed interval, if an observation is left-censored its corresponding entry should be 0.
- `Ri`: the right endpoint of the observed interval, if an observation is right-censored its corresponding entry should be Inf.
- `Ei`: the vector specifying the enrollment times, if no enrollment criteria is used then its corresponding entry should be 0.
- `Xp`: design matrix of predictor variables (in columns), should be specified without an intercept term.
- `n.int`: the number of interior knots to be used.
- `order`: the order of the basis functions.
- `g0`: initial estimate of the spline coefficients; should be of length `n.int+order`.
- `b0`: initial estimate of regression coefficients; should be of length dim(`Xp`)[2].
- `tol`: the convergence criterion of the EM algorithm.

- `t.seq`: an increasing sequence of points at which the cumulative baseline hazard function is evaluated.
- `equal`: logical, if TRUE knots are spaced evenly across the range of the endpoints of the observed intervals and if FALSE knots are placed at quantiles. Defaults to FALSE.

The M-spline and I-spline basis matrices used in this package are generated using two existing R packages (`splines2` and `ICsurv`). Therefore, the users have to install those packages before using `SurvPHspline.EM` function. That is,

```
install.packages(c("splines2","ICsurv"))
library("splines2")
library("ICsurv")
```

For more details about the selection of number of interior knots, order, and starting values, please refer Withana Gamage *et al.* (2022+). The EM algorithm converges when the maximum absolute difference between consecutive parameter updates was less than the specified tolerance (`tol`). The `SurvPHspline.EM()` function output gives the following fields;

- `b`: estimates of the regression coefficients.
- `g`: estimates of the spline coefficients.
- `ll`: the value of the maximized log-likelihood.
- `AIC`: the Akaike information criterion.
- `BIC`: the Bayesian information/Schwarz criterion.
- `bRi`: I-spline basis matrix of dimension c(`n.int+order`, length(`Ri`)).
- `bLi`: I-spline basis matrix of dimension c(`n.int+order`, length(`Li`)).
- `bt`: I-spline basis matrix evaluated at the points `t.seq`.
- `mRi`: M-spline basis matrix of dimension c(`n.int+order`, length(`Ri`)).
- `OPG`: the variance covariance matrix of `b` and `g`.

  NOTE: The computation of the OPG estimator uses the `grad` function in `numDeriv` R package. Therefore, the users have to install `numDeriv` package before using `SurvPHspline.EM` function. That is,

  ```
  install.packages("numDeriv")
  library("numDeriv")
  ```

2. Data Example

Here we provide an example demonstrating the usage of `SurvPHspline.EM` function. The excel file "generated data.csv" provides a data frame with 500 observations on the following 8 variables. The data file is available in the GitHub repository.

- `d1`: Censoring indicator, 1 if failure time was exactly observed, 0 otherwise.
- `d2`: Censoring indicator, 1 if failure time was interval censored, 0 otherwise.
- `d3`: Censoring indicator, 1 if failure time was right censored and, 0 otherwise.
- `Li`: Left endpoint of the observation interval
- `Ri`: Right endpoint of the observation interval
- `Ei`: enrollment time
- `x1`: covariare 1
- `x2`: covariate 2

```
Data<-read.csv(file.choose(),header=TRUE)
# open generated data.csv in the GitHub repository

d1<-Data[,1]
d2<-Data[,2]
d3<-Data[,3]
Li<-Data[,4]
Ri<-Data[,5]
Ei<-Data[,6]
Xp<-as.matrix(Data[,c(7,8)])

# Loading the dependent packages
library(splines2)
library(ICsurv)
library(numDeriv)


library(SurvPHspline)
fit<-SurvPHspline.EM(d1, d2, d3, Li, Ri, Ei, Xp, n.int=1,
                     order=3, g0=rep(1,4) , b0=rep(0,2),
                     tol=.00001, t.seq=seq(0,10,0.1), equal = FALSE)
```

```
fit$b
# [1]  0.2899589 -0.2740518

fit$OPG
```

```
#              [,1]          [,2]          [,3]          [,4]         [,5]         [,6]
# [1,]  0.0242530495  0.0009772677 -0.0012599931 -0.0006347869 -0.032944558  0.033945245
# [2,]  0.0009772677  0.0269306120  0.0003036076 -0.0002903579  0.005201548 -0.008379763
# [3,] -0.0012599931  0.0003036076  0.0013154406 -0.0031899081  0.009380704 -0.017356962
# [4,] -0.0006347869 -0.0002903579 -0.0031899081  0.0167072598 -0.043498155  0.093542136
# [5,] -0.0329445580  0.0052015480  0.0093807037 -0.0434981546  0.225236194 -0.486688557
# [6,]  0.0339452452 -0.0083797632 -0.0173569621  0.0935421355 -0.486688557  2.208201428
```

```
# Baseline survival function
S<-exp(-(fit$bt %*% fit$g))

tseq<-seq(0,10,0.1)
plot(tseq,S,type="l",xlab="",ylab="",cex.lab=0.25)
# x axis
mtext(text = "t",
      side = 1,line = 2,cex=1)

# y axis
mtext(text = expression(S[0](t)),
      side = 2,
      line = 2,cex=1)
```

References

Withana Gamage, P., McMahan, C., and Wang, L. (2022+). *A flexible parametric approach for analyzing arbitrarily censored data that are potentially subject to left truncation under the proportional hazards model.* Submitted.

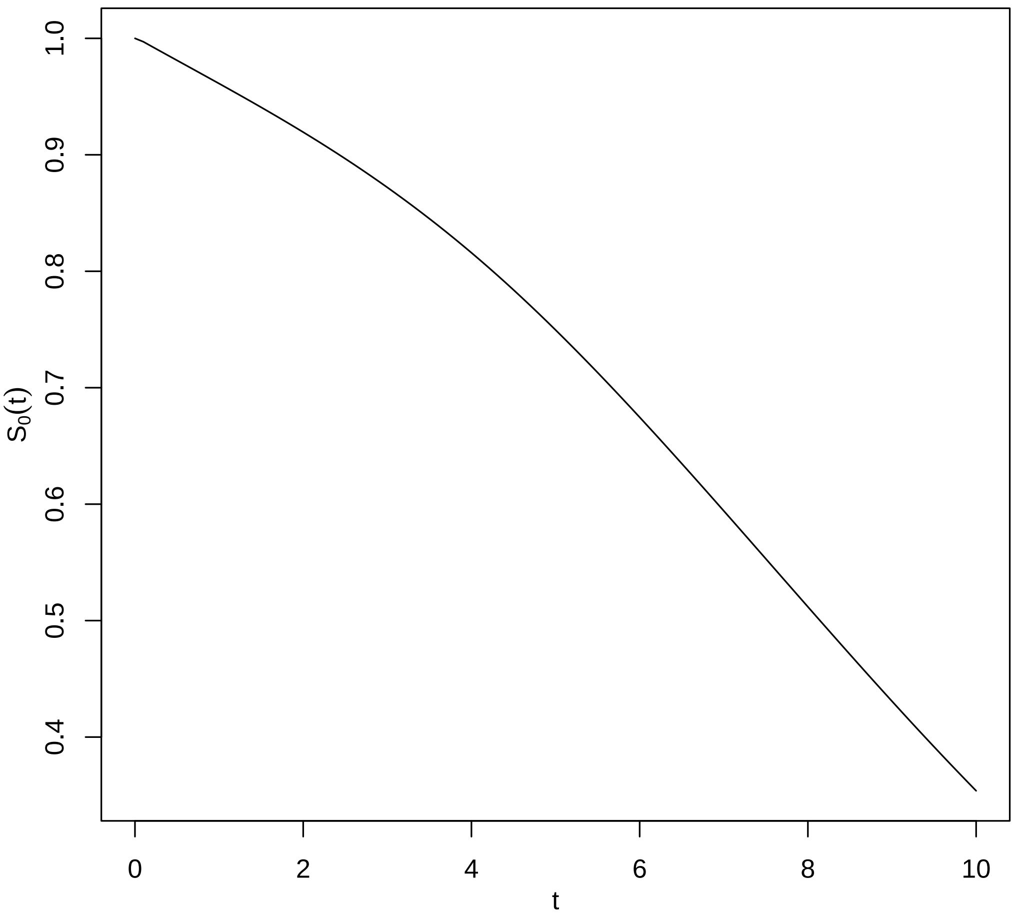

Figure 1: The estimated baseline survival function