Introduction to the rstpm2 package

Mark Clements
Karolinska Institutet

Abstract

This vignette outlines the methods and provides some examples for link-based survival models as implemented in the R **rstpm2** package.

Keywords: survival, splines.

1. Background and theory

Link-based survival models provide a flexible and general approach to modelling survival or time-to-event data. The survival function $S(t|x)$ to time t for covariates x is defined in terms of a link function G and a linear prediction $\eta(t,x)$, such that

$$S(t|x) = G(\eta(t,x))$$

where η is a function of both time t and covariates x. The linear predictor can be constructed in a flexible manner. Royston and Parmar (2003) focused on time being modelled using natural splines for log-time, including left truncation and relative survival. We have implemented the Royston-Parmar model class and extended it in several ways, allowing for: (i) general parametric models for $\eta(t,x)$, including B-splines and natural splines for different transformations of time; (ii) general semi-parametric models for $\eta(t,x)$ including penalised smoothers together with unpenalised parametric functions; (iii) interval censoring; and (iv) frailties using Gamma and log-Normal distributions. Fully parametric models are estimated using maximum likelihood, while the semi-parametric models are estimated using maximum penalised likelihood with smoothing parameters selected using A more detailed theoretical development is available from the paper by Liu, Pawitan and Clements (available on request).

Why would you want to use these models?

2. Mean survival

This has a useful interpretation for causal inference.

$E_Z(S(t|Z,X=1)) - E_Z(S(t|Z,X=0))$

```
fit <- stpm2(...)
predict(fit,type="meansurv",newdata=data)
```

3. Cure models

For cure, we use the melanoma dataset used by Andersson and colleagues for cure models with Stata's stpm2 (see http://www.pauldickman.com/survival/).

Initially, we merge the patient data with the all cause mortality rates.

```
> popmort2 <- transform(rstpm2::popmort,exitage=age,exityear=year,age=NULL,year=NULL)
> colon2 <- within(rstpm2::colon, {
+   status <- ifelse(surv_mm>120.5,1,status)
+   tm <- pmin(surv_mm,120.5)/12
+   exit <- dx+tm*365.25
+   sex <- as.numeric(sex)
+   exitage <- pmin(floor(age+tm),99)
+   exityear <- floor(yydx+tm)
+   ##year8594 <- (year8594=="Diagnosed 85-94")
+ })
> colon2 <- merge(colon2,popmort2)
```

For comparisons, we fit the relative survival model without and with cure.

```
> fit0 <- stpm2(Surv(tm,status %in% 2:3)~I(year8594=="Diagnosed 85-94"),
+              data=colon2,
+              bhazard=colon2$rate, df=5)

> summary(fit <- stpm2(Surv(tm,status %in% 2:3)~I(year8594=="Diagnosed 85-94"),
+                      data=colon2,
+                      bhazard=colon2$rate,
+                      df=5,cure=TRUE))
```

```
Maximum likelihood estimation

Call:
mle2(minuslogl = negll, start = coef, eval.only = TRUE, vecpar = TRUE,
    gr = function (beta)
    {
        localargs <- args
        localargs$init <- beta
        localargs$return_type <- "gradient"
        return(.Call("model_output", localargs, PACKAGE = "rstpm2"))
    }, control = list(parscale = c(`(Intercept)` = 1, `I(year8594 == "Diagnosed 85-94")TRU
    `nsx(log(tm), df = 5, cure = TRUE)1` = 1, `nsx(log(tm), df = 5, cure = TRUE)2` = 1,
    `nsx(log(tm), df = 5, cure = TRUE)3` = 1, `nsx(log(tm), df = 5, cure = TRUE)4` = 1,
    `nsx(log(tm), df = 5, cure = TRUE)5` = 1), maxit = 300),
    lower = -Inf, upper = Inf)

Coefficients:
                                      Estimate Std. Error  z value     Pr(z)
```

```
(Intercept)                               -3.977663   0.054782 -72.6093 < 2.2e-16
I(year8594 == "Diagnosed 85-94")TRUE -0.155511   0.025089  -6.1984 5.704e-10
nsx(log(tm), df = 5, cure = TRUE)1    3.323382   0.053169  62.5058 < 2.2e-16
nsx(log(tm), df = 5, cure = TRUE)2    3.628899   0.053163  68.2597 < 2.2e-16
nsx(log(tm), df = 5, cure = TRUE)3    1.634974   0.022466  72.7752 < 2.2e-16
nsx(log(tm), df = 5, cure = TRUE)4    6.592489   0.111512  59.1192 < 2.2e-16
nsx(log(tm), df = 5, cure = TRUE)5    3.371954   0.042789  78.8036 < 2.2e-16

(Intercept)                          ***
I(year8594 == "Diagnosed 85-94")TRUE ***
nsx(log(tm), df = 5, cure = TRUE)1   ***
nsx(log(tm), df = 5, cure = TRUE)2   ***
nsx(log(tm), df = 5, cure = TRUE)3   ***
nsx(log(tm), df = 5, cure = TRUE)4   ***
nsx(log(tm), df = 5, cure = TRUE)5   ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

-2 log L: 42190.77

> predict(fit,head(colon2),se.fit=TRUE)

   Estimate     lower     upper
1 0.8611043 0.8543119 0.8676050
2 0.7934962 0.7850418 0.8016614
3 0.6967834 0.6863627 0.7069356
4 0.8611043 0.8543119 0.8676050
5 0.8221508 0.8143497 0.8296593
6 0.8611043 0.8543119 0.8676050
```

The estimate for the year parameter from the model without cure is within three significant figures with that in Stata. For the predictions, the Stata model gives:

```
     +-----------------------------------+
     |      surv   surv_lci   surv_uci |
     |-----------------------------------|
  1. | .86108264   .8542898   .8675839 |
  2. | .79346526   .7850106   .8016309 |
  3. | .69674037   .6863196   .7068927 |
  4. | .86108264   .8542898   .8675839 |
  5. | .82212425   .8143227   .8296332 |
     |-----------------------------------|
  6. | .86108264   .8542898   .8675839 |
     +-----------------------------------+
```

We can estimate the proportion of failures prior to the last event time:

```
> newdata.eof <- data.frame(year8594 = unique(colon2$year8594),
+                           tm=10)
> 1-predict(fit0, newdata.eof, type="surv", se.fit=TRUE)
```

```
   Estimate     lower     upper
1 0.6060950 0.6208814 0.5913491
2 0.5512519 0.5658463 0.5367742
```

```
> 1-predict(fit, newdata.eof, type="surv", se.fit=TRUE)
```

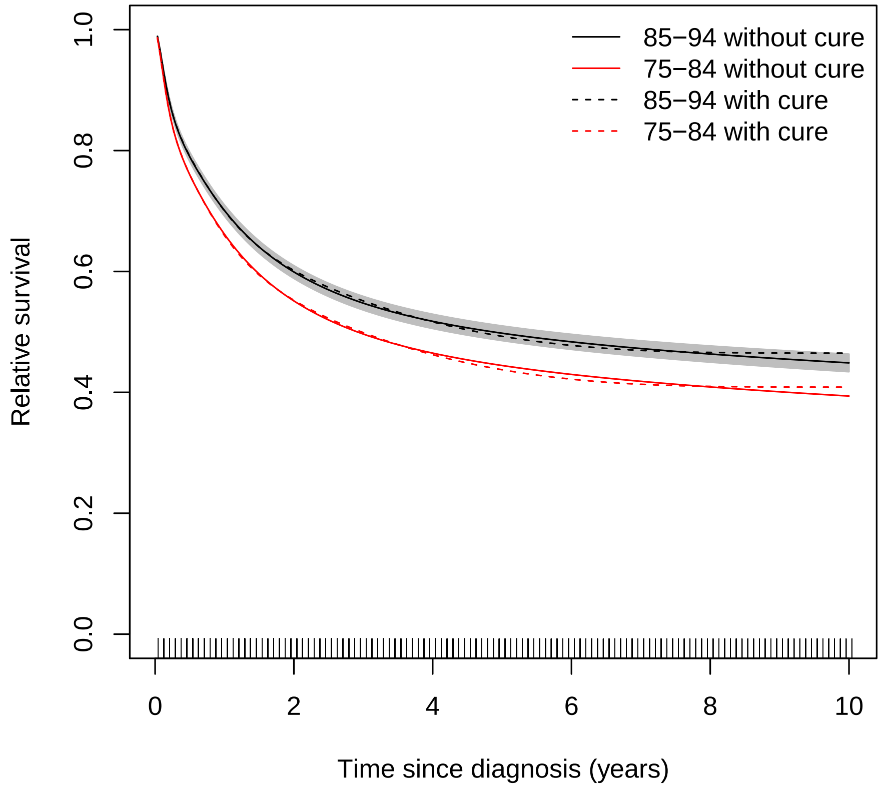
```
   Estimate     lower     upper
1 0.5912976 0.6054691 0.5771835
2 0.5350852 0.5485412 0.5217471
```

```
> predict(fit, newdata.eof, type="haz", se.fit=TRUE)
```

```
      Estimate        lower        upper
1 1.253896e-06 1.092818e-06 1.438717e-06
2 1.073307e-06 9.334234e-07 1.234153e-06
```

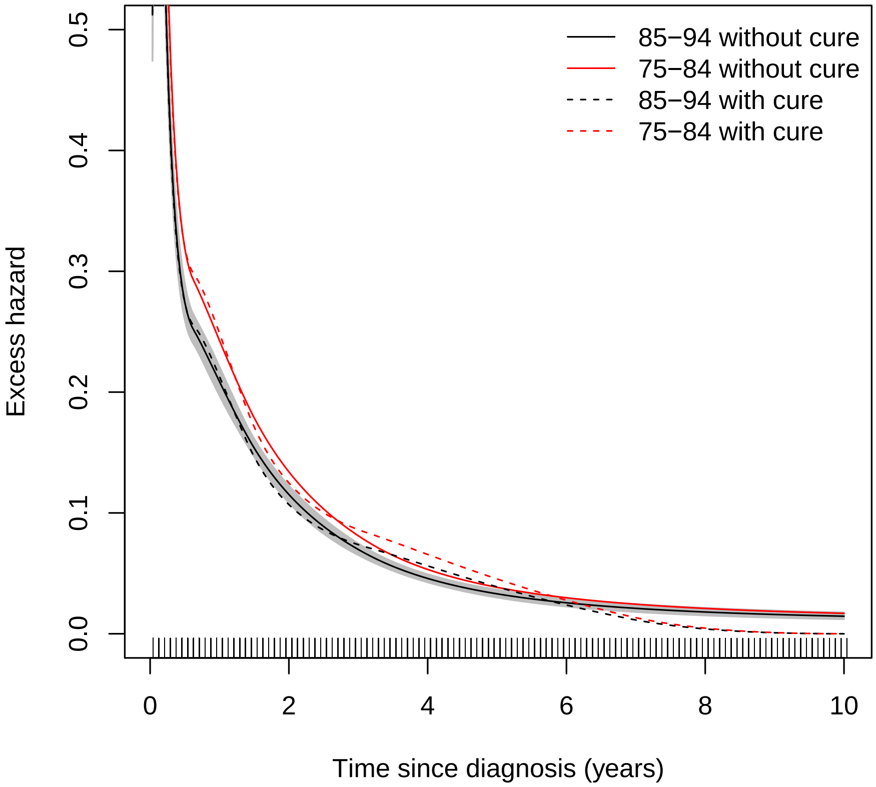
We can plot the predicted survival estimates:

```
> tms=seq(0,10,length=301)[-1]
> plot(fit0,newdata=data.frame(year8594 = "Diagnosed 85-94", tm=tms), ylim=0:1,
+      xlab="Time since diagnosis (years)", ylab="Relative survival")
> plot(fit0,newdata=data.frame(year8594 = "Diagnosed 75-84",tm=tms),
+      add=TRUE,line.col="red",rug=FALSE)
> ## warnings: Predicted hazards less than zero for cure
> plot(fit,newdata=data.frame(year8594 = "Diagnosed 85-94",tm=tms),
+      add=TRUE,ci=FALSE,lty=2,rug=FALSE)
> plot(fit,newdata=data.frame(year8594="Diagnosed 75-84",tm=tms),
+      add=TRUE,rug=FALSE,line.col="red",ci=FALSE,lty=2)
> legend("topright",c("85-94 without cure","75-84 without cure",
+                     "85-94 with cure","75-84 with cure"),
+        col=c(1,2,1,2), lty=c(1,1,2,2), bty="n")
```



And the hazard curves:

```
> plot(fit0,newdata=data.frame(year8594 = "Diagnosed 85-94", tm=tms),
+      ylim=c(0,0.5), type="hazard",
+      xlab="Time since diagnosis (years)",ylab="Excess hazard")
> plot(fit0,newdata=data.frame(year8594 = "Diagnosed 75-84", tm=tms),
+      type="hazard",
+      add=TRUE,line.col="red",rug=FALSE)
> plot(fit,newdata=data.frame(year8594 = "Diagnosed 85-94", tm=tms),
+      type="hazard",
+      add=TRUE,ci=FALSE,lty=2,rug=FALSE)
> plot(fit,newdata=data.frame(year8594="Diagnosed 75-84", tm=tms),
+      type="hazard",
+      add=TRUE,rug=FALSE,line.col="red",ci=FALSE,lty=2)
> legend("topright",c("85-94 without cure","75-84 without cure",
+                     "85-94 with cure","75-84 with cure"),
+        col=c(1,2,1,2), lty=c(1,1,2,2), bty="n")
```



Affiliation:

Mark Clements
Department of Medical Epidemiology and Biostatistics
Karolinska Institutet
Email: mark.clements@ki.se