C.2 Modern Survival Problem sheet 2: Nonparametric estimation of survival curves

(1) Consider a situation where the multiplicative intensity model holds, and there is unobserved right censoring. That is, for some individuals we observe the event time T_i and $\delta_i = 1$; for others, we observe $\delta_i = 0$ and no event time. Suppose the right censoring is independent of event times, all n individuals are independent, and the distribution of censoring times is known to have cdf G. (So $G(c)$ is the probability of being censored before time c.) Let $t_1 < t_2 < \cdots < t_k$ be the observed event times (assumed distinct).

Show that

$$\hat{A}(t) = \sum_{t_i \le t} \Big((n - i + 1)\big(1 - G(t_i)\big) \Big)^{-1}$$

is an unbiased estimator for the cumulative hazard, and derive an estimator for the variance.

Let $\mathcal{F}_t$ be the σ-algebra generated by the events up to time t, and $\mathcal{G}_t$ the σ-algebra generated by events and censoring times up to time t.

We know that the counting process $N(t)$ — the number of events at times $\le t$ — has $\mathcal{G}_t$-compensator

$$\Lambda(t) = \int_0^t Y(s) dA(s).$$

Thus, the $\mathcal{F}_t$-compensator, by the Innovation Theorem, is

$$\widetilde{\Lambda}(t) := \mathbb{E}\big[\Lambda(t) \,\big|\, \mathcal{F}_t\big] = \int_0^t \mathbb{E}\big[Y(s) \,\big|\, \mathcal{F}_t\big] dA(s)$$

(since $A(s)$ is deterministic).

Conditioned on $\mathcal{F}_t$, which includes only the times of the events, the probability that an individual is still at risk at time s is 0 if they have already had their event by time s, and $(1 - G(s))$ if they have not. There are $n - N(s-)$ individuals who have not yet had an event at time s, so

$$\widetilde{\Lambda}(t) = \int_0^t \big(n - N(s-)\big)\big(1 - G(s)\big) dA(s).$$

Thus $M(t) := N(t) - \widetilde{\Lambda}(t)$ is an $\mathcal{F}_t$-martingale. Since $\big(n - N(s-)\big)\big(1 - G(s)\big)$ is predictable, it follows that

$$\begin{aligned}
\widetilde{M}(t) :&= \int_0^t \big(n - N(s-)\big)^{-1}\big(1 - G(s)\big)^{-1} dM(s) \\
&= \sum_{t_i \le t} \big(n - N(t_i-)\big)\big(1 - G(t_i)\big)^{-1} - A(t) \\
&= \hat{A}(t) - A(t)
\end{aligned}$$

is also a martingale. Thus, its expectation is 0.

The optional variation is

$$\sum_{t_i \le t} \big(n - i + 1\big)^{-2} \big(1 - G(t_i)\big)^{-2},$$

which may thus serve as an unbiased estimator for the variance of $\hat{A}(t)$.

Note that this estimator does not use all of the data. We could improve our estimation by using the filtration $\mathcal{F}?_t : \mathcal{F}_t \vee \langle \delta_i : i = 1, \dots, n \rangle$. That is, we include at all times the information about who has ultimately been censored. Conditioned on $\mathcal{F}?_t$ we know that there are $C := n - \sum \delta_i$ individuals who will ultimately be censored. Thus it makes sense to write

$$Y(s) = \sum_{i:\delta_i = 1} \mathbf{1}_{\{T_i > s\}} + \sum_{i:\delta_i = 0} \mathbf{1}_{\{C_i > s\}},$$

where C_i is the (unobserved) censoring time for individual i. The only variables that are not in $\mathcal{F}?_t$ are the C_i. Thus, for $s \le t$,

$$\begin{aligned} \mathbb{E}\big[Y(s) \,\big|\, \mathcal{F}?_t\big] &= \sum_{i:\delta_i = 1} \mathbf{1}_{\{T_i > s\}} + \sum_{i:\delta_i = 0} \mathbb{P}\big\{C_i > s \,\big|\, \delta_i = 0\big\} \\ &= \big(n - N(s-) - C\big) - C\mathbb{P}\big\{C_i > s \,\big|\, T_i > C_i\big\}. \end{aligned}$$

In fact (and contrary to what I somewhat glibly claimed in the lecture), it?s not straightforward to turn this into an estimator for A!

(2) Nonparametric estimators are inevitably less efficient (that is, have larger errors, on average) than parametric estimators. Consider the case when n individuals are observed up to time t. Their event times are independent and exponentially distributed with unknown parameter λ, and we observe all event times. We wish to compare two different possible estimators for the cumulative hazard up to time t: First, taking advantage of the knowledge that the data come from an exponential distribution; and second, using the nonparametric Nelson–Aalen estimator.

(a) i. Show that the MLE for $\Lambda(t)$ under the exponential model is

$$\frac{nt}{\sum_{i=1}^n T_i}.$$

The log likelihood is

$$\ell(\lambda) = n \log \lambda - \lambda \sum T_i.$$

Setting the derivative to 0 and solving for λ we get $\hat{\lambda} = n / \sum T_i$. The result follows, since $\Lambda(t) = \lambda t$.

ii. Compute the (approximate) variance for this estimator.

We know that $G := \sum_{i=1}^{n} T_i$ has Gamma distribution with parameters (n, λ), so has density

$$g(x) = \frac{\lambda^n}{(n-1)!} x^{n-1} e^{-\lambda x}.$$

Thus

$$\mathbb{E}[\hat{\lambda}] = n \int_0^\infty \frac{\lambda^n}{(n-1)!} x^{n-2} e^{-\lambda x} = \frac{n\lambda}{n-1},$$
$$\mathbb{E}[\hat{\lambda}^2] = n^2 \int_0^\infty \frac{\lambda^n}{(n-1)!} x^{n-2} e^{-\lambda x} = \frac{n^2\lambda^2}{(n-1)(n-2)},$$
$$\mathrm{Var}(\hat{\lambda}) = \frac{n^2\lambda^2}{(n-1)^2(n-2)} \approx \frac{\lambda^2}{n}.$$

Thus $\mathrm{Var}(\hat{\lambda}) \approx \lambda^2 t^2/n$.

The mean squared error (which is what we really should be looking at) adds the squared bias to this, which is $\lambda^2/(n-1)^2$, but this doesn't change anything significant for large n.

(b) Using the inequality

$$\log n + \gamma \le \sum_{i=1}^{n} \frac{1}{i} \le \log n + \gamma + \frac{1}{2n},$$

show that the Nelson–Aalen estimator for $S(t)$ is approximately $Y(t+)/n$ (the empirical fraction surviving to time n), and find a bound for the error — that is, for the maximum difference between $\widetilde{S}(s)$ and $Y(s+)/n$ on $0 \le s \le t$.

We have

$$-\log \widetilde{S}(s) = \sum_{j=1}^{N(s)} \frac{1}{n+1-j}.$$

Since $Y(s+) = n + 1 - N(s)$,

$$\left|\log \widetilde{S}(s) - \log \frac{Y(s+)}{n}\right| \le \frac{1}{2Y(s+)}.$$

Using the general inequality $|e^{-x} - e^{-y}| \le |x - y|$ for any $x, y \ge 0$, we conclude that for all $0 \le s \le t$,

$$\left|\widetilde{S}(s) - \frac{Y(s+)}{n}\right| \le e^{1/2Y(s+)}.$$

We can improve this, if we wish, to $e^{1/2Y(s+)} \cdot \frac{Y(s+)}{n-1}$.

(c) Use this to estimate the variance of $\hat{A}(t)$, the Nelson–Aalen estimator. Show that this variance is larger than the variance for the parametric estimator above.
Writing $\hat{A}(t) = -\log \widetilde{S}(t) = -\log Y(t+)/n$, we have

$$\left(\hat{A}(t) - \Lambda(t)\right)^2 \approx \log^2\left(\frac{\widetilde{S}(t)}{S(t)}\right) \qquad = \log^2\left(1 + \frac{\widetilde{S}(t) - S(t)}{S(t)}\right)$$
$$\approx \left(\frac{\widetilde{S}(t) - S(t)}{S(t)}\right)^2 + O\left(|\widetilde{S}(t)/S(t) - 1|^3\right),$$

so

$$\operatorname{Var}(\hat{A}(t)) \approx S(t)^{-2} \operatorname{Var}(\widetilde{S}(t)) = n^{-2} S(t)^{-2} \operatorname{Var}(Y(t+)).$$

Since $Y(t+)$ has binomial distribution with parameters $(n, S(t))$, its variance is $nS(t)(1 - S(t))$, so we have

$$\operatorname{Var}(\hat{A}(t)) \approx n^{-1}\left(\frac{1}{S(t)} - 1\right) = n^{-1}\left(e^{\lambda t} - 1\right).$$

(d) Show that this variance is larger than the variance for the parametric estimator above.
We note that for any $u > 0$ we have

$$\begin{aligned} e^u - 1 - u^2 &= u - \frac{u^2}{2} + \sum_{i=3}^{\infty} \frac{u^i}{i!} \\ &> \sum_{i=1}^{\infty} (-1)^{i-1} \frac{u^i}{i!} \\ &= 1 - e^{-u} \\ &> 0. \end{aligned}$$

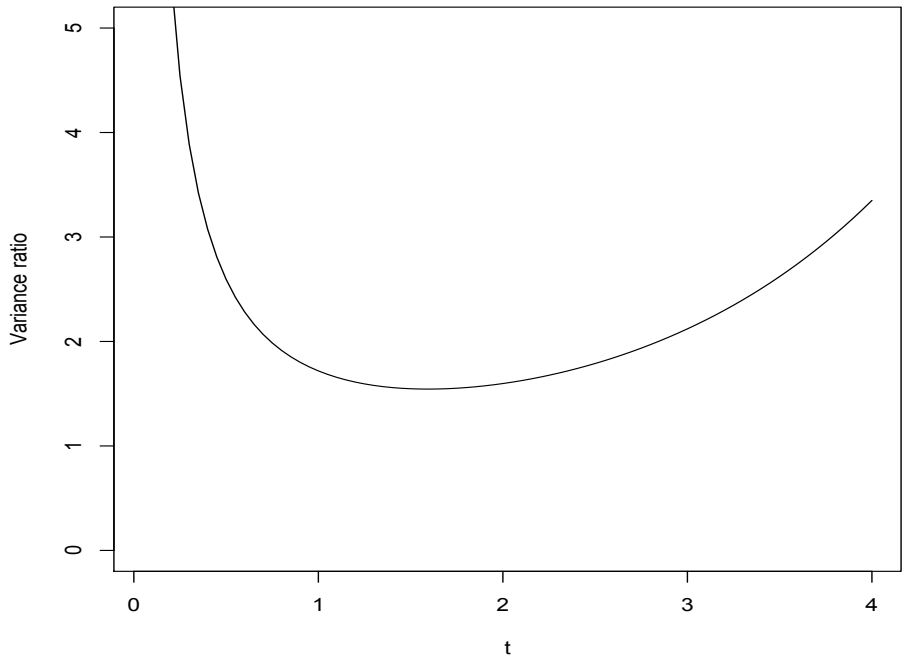
So $e^{\lambda t} - 1 > (\lambda t)^2$. The parametric estimator has variance approximately $n^{-1}(\lambda t)^2$, while the nonparametric estimator has variance approximately $n^{-1}(e^{\lambda t} - 1)$, so the nonparametric variance is larger.

(e) Plot the ratio of the variances for a range of values of t, for the case $\lambda = 1$ and $n = 1000$.

(f) Why might one prefer to use the nonparametric estimator, even when there is no censoring?
If the exponential model is not actually a good fit, then the $\hat{\Lambda}(t)$ obtained from it will be distorted.

(3) What is the intensity λ of the survival process at time t?
The intensity of the process is equal to $\alpha \times$ (number of subjects at risk at time t), or

$$\lambda(t) = \alpha(n - N(t-)).$$



(a) R code runs faster if you replace loops with vector operations. Can you get rid of the loop in this code?

(b) Modify the code to plot the martingale $N(t) - \int_0^t \lambda(s)ds$.

(c) Add a routine that computes the optional variation process. Run a simulation and plot it.

(d) Modify the program to apply to $\alpha(t) = 2t$.

```
################################################
############ Original code #####################
################################################
n=50
alpha=1
taus=NULL  #Accumulates the inter-event times

set.seed(0)# this can be used to validate similarity of the output
for (i in 1:n){
        taus=c(taus,rexp(1,alpha*(n+1-i))) # The next inter-event time
```

```
}

T=cumsum(taus)  # Turn the inter-event times into event times


# First create an empty set of axes.
# y ranges from n down to 0, x ranges from 0 up to max(T)

# Make an empty set of axes
plot(NULL,NULL,xlim=c(0,max(T)),
    ylim=c(0,n),
    xlab='Time',
    ylab='Number of events')

# Now plot flat lines for all the segments between arrivals

segments(c(0,T[1:n]),0:n,c(T[1:n],T[n]+1),0:n)
# T[n]+1 is there to extend the final interval one
# time unit from the last time
# In principle it extends forever

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

################################################
###############   (3 a )    ###################
################################################



n=50
alpha=1
taus=NULL  #Accumulates the inter-event times

set.seed(0) #this can be used to validate similarity of the output
# It should produce the same taus as the for loop
##~~~~~ replacing the loop~~~~~~
taus=rexp(n,alpha*( n+1 - c(1:n) )) # The next inter-event time
```

```
################################################
###############   (3 b )    ####################
################################################

# Lambda(t) is equal to t(alpha(n-N(t))

M <- c(1:50) - T*( alpha*(n +1 -c(1:n) ) )

plot(NULL,NULL,xlim=c(0,max(T)),
     ylim=c(min(M), max(M) ),
     xlab='Time',ylab='M(t)')
segments(c(0,T[1:n]), M ,c(T[1:n],T[n]+1) , M )


################################################
###############   (3 c )    ####################
################################################

# The optional variation process is just sum( Y_i^2 )
# where Y_i is the changes in the process

optvar <- cumsum( ( 1:n - 0:(n-1) )^2 )

optvar <- c(0,optvar)
#Plotting it:

plot(NULL,NULL,xlim=c(0,max(T)),
      ylim=c(0,n),xlab='Time',ylab='Optional Variation process')
segments(c(0,T[1:n]), optvar ,c(T[1:n],T[n]+1), optvar )


################################################
###############   (3 d )    ####################
################################################

# Now that we have a more complex hazard rate,
# we will have to use what we learned in question 4

#Treating N(t) as observed:
#lambda(t)    = 2t  * (n - N(t))
```

```r
#Lambda(t)     = t^2 * (n-N(t))
#Lambda^-1(t) = sqrt( t/ (n-N(t)) )

n <- 50

Lambda_inv <- function(x , N )  sqrt( x / (n + 1 - N) )

taus= Lambda_inv( rexp(50,1)  , 1:50 )


T=cumsum(taus)  # Turn the inter-event times into event times

# First create an empty set of axes.
# y ranges from n down to 0, x ranges from 0 up to max(T)

# Make an empty set of axes
plot(NULL,NULL,xlim=c(0,max(T)),
      ylim=c(0,n),
      xlab='Time',
      ylab='Number of events')

# Now plot flat lines for all the segments between arrivals

segments(c(0,T[1:n]),0:n,c(T[1:n],T[n]+1),0:n)
# T[n]+1 is there to extend the final interval
# one time unit from the last time
# In principle it extends forever


## Plotting the martingale
Lambda <- function( t , N  ) (t^2) * (n + 1 - N )

M <- c(1:50) - Lambda(T, 1:50)

plot(NULL,NULL,xlim=c(0,max(T)),
              ylim=c(min(M), max(M) ),
              xlab='Time',ylab='M(t)')
segments(c(0,T[1:n]), M ,c(T[1:n],T[n]+1) , M )
```

```
## Plotting the optional variation process
# The optional variation process is just sum( Y_i^2 )
# where Y_i is the changes in the process

optvar <- cumsum( ( 1:n - 0:(n-1) )^2 )

optvar <- c(0,optvar)
#Plotting it:

plot(NULL,NULL,xlim=c(0,max(T)),ylim=c(0,n),xlab='Time',
     ylab='Optional Variation process')
segments(c(0,T[1:n]), optvar ,c(T[1:n],T[n]+1), optvar )
```

(4) The data set `ovarian`, included in the `survival` package, presents data for 26 ovarian cancer patients, receiving one of two treatments, which we will refer to as the *single* and *double* treatments.

(a) Create a survival object for the times in this database.

(b) Compute and plot the Kaplan–Meier estimator for the survival curves. (For a small extra challenge, plot the single-treatment survival curve black, and the double-treatment curve red.) You may use the `survfit` function.

```
library(survival)

## a ##

surv_object <- Surv(ovarian$futime, ovarian$fustat)

# To have a look at what has been computed about survival


## b ##
plot(survfit(surv_object~ovarian$rx), main="Kaplan-Meier")

> summary(surv_object)
Call: survfit(formula = Surv(futime, fustat) ~ rx)

#                rx=1
# time n.risk n.event survival std.err lower 95\% CI upper 95\% CI
 #  59     13       1    0.923  0.0739        0.789        1.000
 # 115     12       1    0.846  0.1001        0.671        1.000
 # 156     11       1    0.769  0.1169        0.571        1.000
```

```
#  268     10       1    0.692  0.1280        0.482        0.995
 # 329      9       1    0.615  0.1349        0.400        0.946
#  431      8       1    0.538  0.1383        0.326        0.891
#  638      5       1    0.431  0.1467        0.221        0.840
#
 #               rx=2
# time n.risk n.event survival std.err lower 95\% CI upper 95\% CI
#  353     13       1    0.923  0.0739        0.789        1.000
#  365     12       1    0.846  0.1001        0.671        1.000
#  464      9       1    0.752  0.1256        0.542        1.000
#  475      8       1    0.658  0.1407        0.433        1.000
#  563      7       1    0.564  0.1488        0.336        0.946

#for extra challenge:
plot(survfit(surv_object~ovarian$rx) ,
      col=c("black","red"),
      main="Kaplan-Meier")
legend("bottomright",
      c( "single-treatment", "double-treatment"),
      col=c("black","red") , lty=1 )

## c ##

plot(survfit(surv_object~ovarian$rx, type='fleming-harrington'),main="Nelson-Aalen")


### The rest is to do this more 'by hand', computing the relevant quantities
### and directly computing the Nelson-Aalen estimator.

attach(ovarian)

x=order(futime)
futime=futime[x]
fustat=fustat[x]
rx=rx[x]

ns=rev(cumsum(rev(rx==1)))
nd=rev(cumsum(rev(rx==2)))
hs=round(fustat*(rx==1)/ns,2)
hd=round(fustat*(rx==2)/nd,2)
```

```
NelsonAalenTable =
  subset(data.frame(t_i=futime,  n_single=ns, n_double=nd,
h_single=hs,h_double=hd,A_single=cumsum(hs),A_double=cumsum(hd)), h_single+h_double>0)

> NelsonAalenTable
   t_i n_single n_double h_single h_double A_single A_double vars vard
1   59       13       13     0.08     0.00     0.08     0.00 0.01 0.00
2  115       12       13     0.08     0.00     0.16     0.00 0.01 0.00
3  156       11       13     0.09     0.00     0.25     0.00 0.02 0.00
4  268       10       13     0.10     0.00     0.35     0.00 0.03 0.00
5  329        9       13     0.11     0.00     0.46     0.00 0.04 0.00
6  353        8       13     0.00     0.08     0.46     0.08 0.04 0.01
7  365        8       12     0.00     0.08     0.46     0.16 0.04 0.01
10 431        8        9     0.12     0.00     0.58     0.16 0.06 0.01
12 464        6        9     0.00     0.11     0.58     0.27 0.06 0.03
13 475        6        8     0.00     0.12     0.58     0.39 0.06 0.04
15 563        5        7     0.00     0.14     0.58     0.53 0.06 0.06
16 638        5        6     0.20     0.00     0.78     0.53 0.10 0.06
```