Advanced Statistical Methods Hw7

Do Hyup Shin

2021-11-09

Problem 9.3

Redraw Figure 9.2, changing the "knot" location from 11 to 12.

Solution

First, we will check case Arm_A of the NCOG. In the book, n = number at risk(patients), y = number of deaths, l = lost to followup, h = hazard rate y/n; $\hat{S}$ = life table survival esimate. Next, we assume that $y_i \overset{indep}{\sim} B(n_i, h_i) \quad \forall i = 1, 2, \ldots, 47$. We will use a generalized linear model. Let $\mu_i = E(y_i) = n_i h_i \quad \forall i = 1, 2, \ldots, 47$. The pdf of y_i is

$$\begin{aligned} f(y_i) &= \binom{n_i}{y_i} h_i^{y_i}(1-h_i)^{n_i - y_i} = exp(y_i log\frac{h_i}{1-h_i} + n_i log(1-h_i) + log\binom{n_i}{y_i}) \\ &= exp(y_i log\frac{\mu_i}{n_i - \mu_i} + n_i log(\frac{n_i - \mu_i}{n_i}) + log\binom{n_i}{y_i}) \\ &= exp(y_i\theta_i - n_i log(1+e^{\theta_i}) + log\binom{n_i}{y_i}) \quad \text{where } \theta_i = log\frac{\mu_i}{n_i - \mu_i} \end{aligned}$$

Define $b(\theta_i) = n_i log(1+e^{\theta_i}), \quad a(\phi) = 1, \quad g(\mu_i) = \theta_i = log\frac{\mu_i}{n_i - \mu_i} = \lambda_i = x_i^T\alpha$ where $x_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}) = (1, i, (i-12)^2\_, (i-12)^3\_)^T, \quad \alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)^T$ and we can express $\mu_i = n_i\frac{e^{\theta_i}}{1+e^{\theta_i}}$. By using this form, calculate log likelihood function $l(\alpha)$.

$$\begin{aligned} l(\alpha) &= \sum_{i=1}^{47}(y_i\theta_i - b(\theta_i) + log\binom{n_i}{y_i}) \\ &= \sum_{i=1}^{47}(y_i x_i^T\alpha - n_i log(1+e^{x_i^T\alpha}) + log\binom{n_i}{y_i}) \end{aligned}$$

Let $S(\alpha)$ be score function.

$$\begin{aligned} S(\alpha_j) &= \frac{\partial S}{\partial \alpha_j} \\ &= \sum_{i=1}^{47}(y_i x_{ij} - n_i\frac{e^{x_i^T\alpha}}{1+e^{x_i^T\alpha}}x_{ij}) \\ &= \sum_{i=1}^{47}(y_i - \mu_i)x_{ij} = 0 \quad \forall j = 1, 2, 3, 4 \end{aligned}$$

We will find the mle of α_j satisfying $S(\alpha_j) = 0$. But, we cannot find the exact solution. So we will use the Iteratively Rewighted least square algorithm method.

Let $g(y_i) \approx z_i = g(\mu_i) + g'(\mu_i)(y_i - \mu_i) = \lambda_i + \frac{d\lambda_i}{d\mu_i}(y_i - \mu_i)$ with $var(z_i) = \frac{var(y_i)}{(\frac{d\mu_i}{d\lambda_i})^2}$. Then, $g'(\mu_i) = \frac{n_i}{\mu_i(n_i-\mu_i)}$ and define $D = diag(\frac{\partial \mu_i}{\partial \lambda_i}) = diag(\frac{\mu_i(n_i-\mu_i)}{n_i})$ and $V = diag(var(y_i)) = diag(\mu_i(\frac{n_i-\mu_i}{n_i}))$.

The IRLS algorithm works as follows.

Step1 Define the kth vector value $\alpha_{(k)}$.

Step2 By using $\alpha_{(k)}$, Calculate $\lambda_{(k)}, \mu_{(k)}, z_{(k)}, D_{(k)}, V_{(k)}$.

Step3 Calculate $\alpha_{(k+1)} = (X^T W_{(k)} X)^{-1} X^T W_{(k)} z_{(k)}$.

Step4 Continue above process until the $\alpha_{(k)}$ converges.

Actually, we can simplify $W = DV^{-1}D = diag(\frac{\mu_i(n_i-\mu_i)}{n_i})diag(\frac{n_i}{\mu_i(n_i-\mu_i)})diag(\frac{\mu_i(n_i-\mu_i)}{n_i}) = diag(\frac{\mu_i(n_i-\mu_i)}{n_i}) = D$. So, we can find the numerical solution of the mle α.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

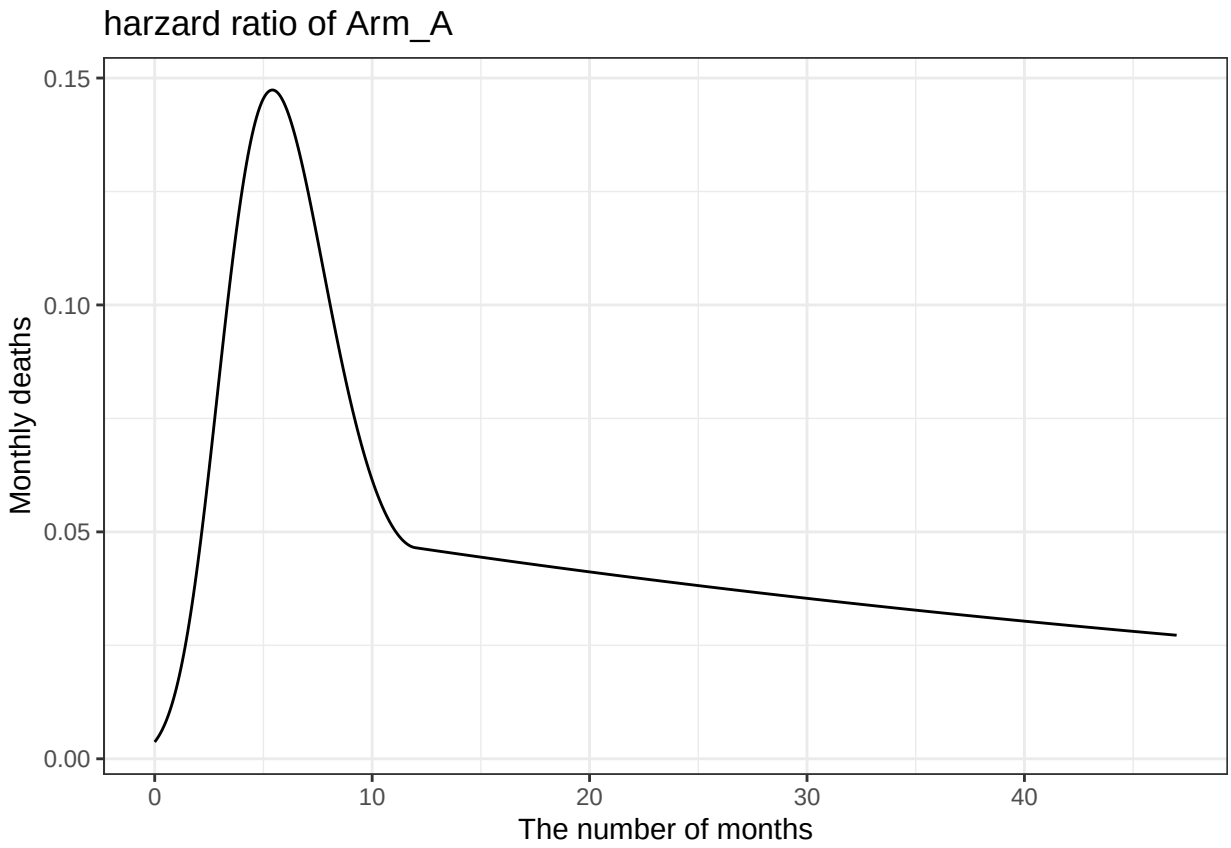
```
library(ggplot2)
NCOG_data <- read.csv("https://web.stanford.edu/~hastie/CASI_files/DATA/ncog.txt", sep = " ")

#Separate A and B
Arm_A <- subset(NCOG_data,subset = arm == "A")
Arm_B <- subset(NCOG_data,subset = arm == "B")

## harzard function of Arm_A
num_A = nrow(Arm_A)
Km_A <- matrix(0, 47, 4)
copy_A = Arm_A

#make the 47 * 4 matrix
for(i in 1:47){
  Km_A[i,1] = nrow(copy_A)
  inter_A = copy_A %>% filter(t > 30.4*(i-1) & t <= 30.4*i)
  Km_A[i, 2] <- length(which(inter_A$d == 1))
  Km_A[i, 3] <- length(which(inter_A$d == 0))
  Km_A[i, 4] <- Km_A[i, 2] / Km_A[i,1]
  copy_A = copy_A %>% filter(t > 30.4*i)
}

x_a <- matrix(0, 47, 4)
for(i in 1:47){
  if(i <= 11){
    x_a[i,] <- c(1, i, (i-12)^2, (i-12)^3)
```

```
  }
  else{
    x_a[i,1:2] <- c(1, i)
  }
}

#number of patients
n_a = Km_A[,1]

#number of deaths
y_a = Km_A[,2]

## Iteratively Rewighted least square(IRLS)
# initial value of alpha
alpha_0 = c(-1, -0.01, 0.1, 0.01)
alpha = matrix(0, 1001, 4)
alpha[1,] = alpha_0

#iterate 1000 times
for(i in 1:1000){

  lambda = x_a %*% alpha[i,]

  # mu_k = n * exp(lambda_k)/ (1 + exp(lambda_k))
  mu = n_a * exp(lambda) / (1 + exp(lambda))

  # z_k = lambda_k + n / (mu_k * (n - mu_k)) * (y - mu_k)
  z = lambda  + (n_a / (mu * (n_a - mu)) ) * (y_a - mu)


  # D_k = diag(mu_k(n - mu_k) / n)
  D = diag(as.numeric(mu*(n_a - mu)/n_a))


  # V_k = diag(mu_k (1 - mu_k/n))
  V = diag(as.numeric(mu* (1 - mu/n_a)))
  # W_k = D %*% solve(V) %*% D

  W = D

  # alpha_(k+1) = (X^t W_k X)^(-1) X^t W_k Z_k
  alpha[i+1, ] = solve((t(x_a) %*% W %*% x_a)) %*% t(x_a) %*% W %*% z
}
alpha_hat_a = alpha[1001,]

#The numerical solution of mle
alpha_hat_a
```

```
## [1] -2.830345147 -0.015866307  0.082670750  0.008488984
```

Thus, the numerical mle of α in Arm_A case is $\hat{\alpha} = (-2.8303, -0.01587, 0.08267, 0.0085)^T$. Next, we'll draw the graph of harzard ratio.

```r
#harzard ratio
harzard_func <- function(x){
  h = 1/(1 + exp(-x%*%alpha_hat_a))
  return(h)
}

#cubic linear spline function
spline_func <- function(x){
  if(x<= 12){
    return(c(1, x, (x-12)^2,(x-12)^3))
  }
  else{
    return(c(1, x, 0, 0))
  }
}

#range of x
range_xa = seq(0, 47, 0.01)
h_a = rep(0, 4701)
for(i in 1:4701){
  v = spline_func(range_xa[i])
  h_a[i] = harzard_func(v)
}

harzard_ratio_a = as.data.frame(cbind(range_xa, h_a))
ggplot(data=harzard_ratio_a, aes(x=range_xa, y=h_a)) + geom_line(color='black', lwd=0.5) + ggtitle("har
xlab("The number of months") + ylab("Monthly deaths") +theme_bw()
```



In the same way as above, the case of Arm_B of the NCOG can also be obtained.

```
## harzard function of Arm_B
num_B = nrow(Arm_B)
Km_B <- matrix(0, 76, 4)
copy_B = Arm_B
for(i in 1:76){
  Km_B[i,1] = nrow(copy_B)
  inter_B = copy_B %>% filter(t > 30.4*(i-1) & t <= 30.4*i)
  Km_B[i, 2] <- length(which(inter_B$d == 1))
  Km_B[i, 3] <- length(which(inter_B$d == 0))
  Km_B[i, 4] <- Km_B[i, 2] / Km_B[i,1]
  copy_B = copy_B %>% filter(t > 30.4*i)
}
x_b <- matrix(0, 76, 4)
for(i in 1:76){
  if(i <= 11){
    x_b[i,] <- c(1, i, (i-12)^2, (i-12)^3)
  }
  else{
    x_b[i,1:2] <- c(1, i)
  }
}
n_b = Km_B[,1]
y_b = Km_B[,2]
## Iteratively Rewighted least square(IRLS)
# initial value of alpha
```

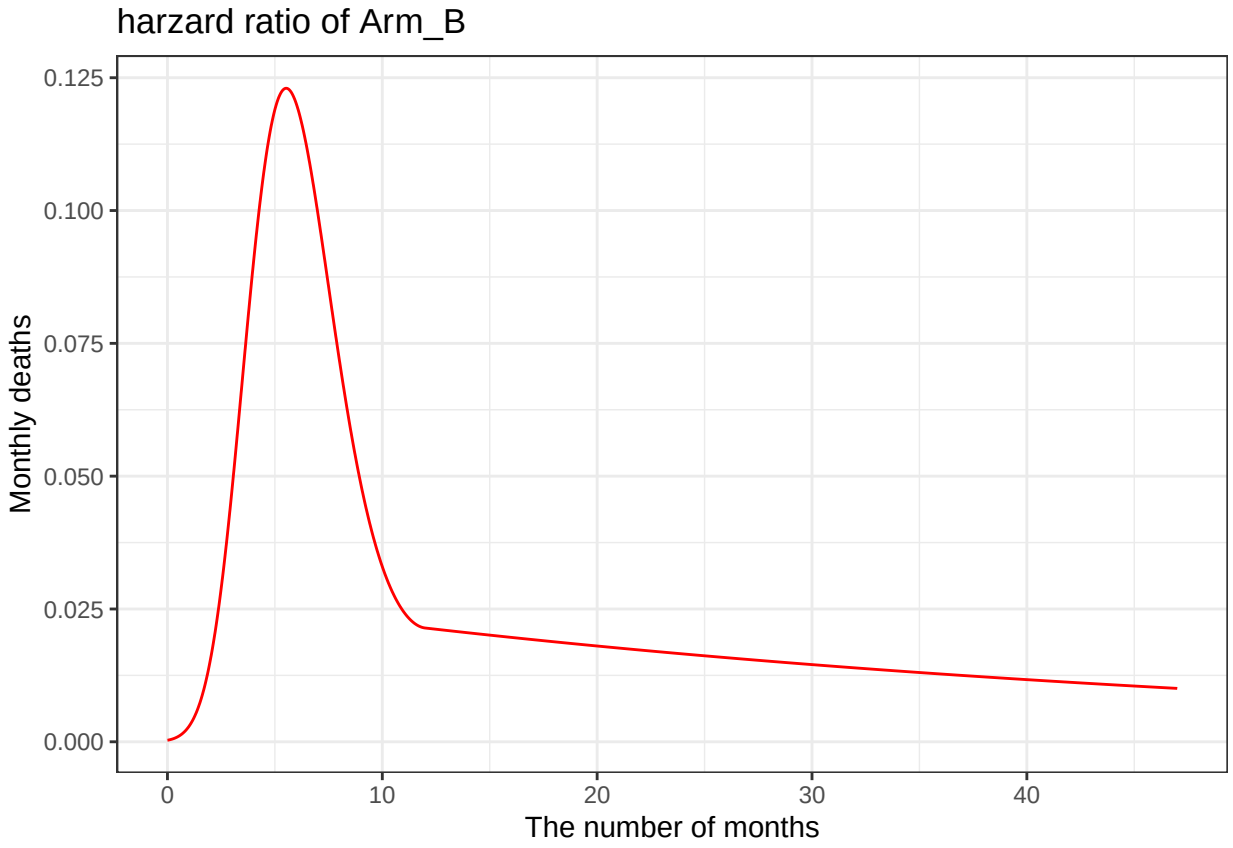
```
alpha_0 = c(1, 0.01, 0.01, 0.01)
alpha = matrix(0, 1001, 4)
alpha[1,] = alpha_0
for(i in 1:1000){
  lambda = x_b %*% alpha[i,]

  # mu_k = n * exp(lambda_k)/ (1 + exp(lambda_k))
  mu = n_b * exp(lambda) / (1 + exp(lambda))

  # z_k = lambda_k + n / (mu_k * (n - mu_k)) * (y - mu_k)
  z = lambda  + (n_b/ (mu * (n_b - mu)) ) * (y_b - mu)


  # D_k = diag(mu_k(n - mu_k) / n)
  D = diag(as.numeric(mu*(n_b-mu)/n_b))


  # V_k = diag(mu_k (1 - mu_k/n))
  V = diag(as.numeric(mu* (1 - mu/n_b)))

  # W_k = D %*% solve(V) %*% D

  W = D

  # alpha_(k+1) = (X^t W_k X)^(-1) X^t W_k Z_k
  alpha[i+1, ] = solve((t(x_b) %*% W %*% x_b)) %*% t(x_b) %*% W %*% z
}
alpha_hat_b = alpha[1001,]

alpha_hat_b
```

```
## [1] -3.55862953 -0.02194139  0.12619500  0.01316970
```

Thus, the numerical mle of α in Arm_B case is $\hat{\alpha} = (-3.5586, -0.0219, 0.1262, 0.0132)^T$. Next, we'll draw the graph of harzard ratio.

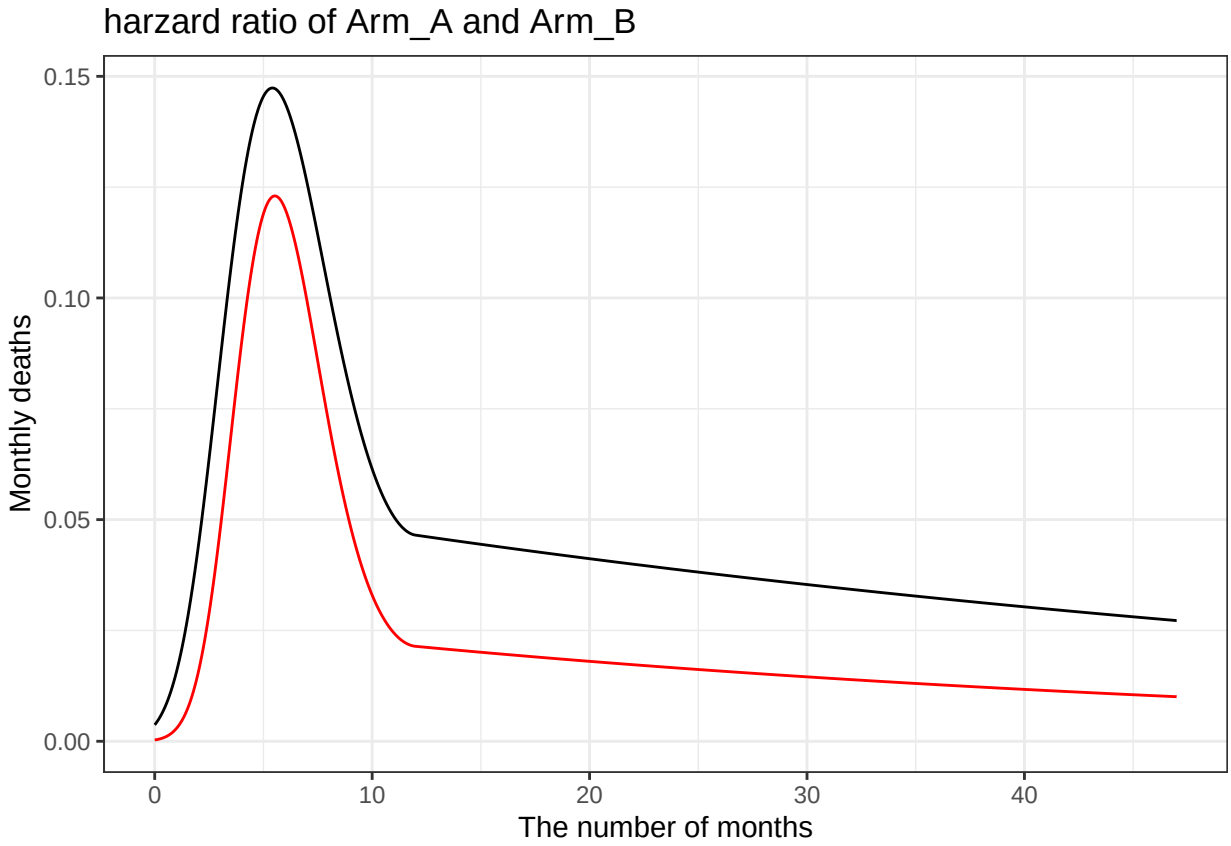
```
harzard_func <- function(x){
  h = 1/(1 + exp(-x%*%alpha_hat_b))
  return(h)
}
spline_func <- function(x){
  if(x<= 12){
    return(c(1, x, (x-12)^2,(x-12)^3))
  }
  else{
    return(c(1, x, 0, 0))
  }
}
range_xb = seq(0, 47, 0.01)
h_b = rep(0, 4701)
for(i in 1:4701){
  v = spline_func(range_xb[i])
  h_b[i] = harzard_func(v)
}
harzard_ratio_b = as.data.frame(cbind(range_xb, h_b))
```

```
ggplot(data=harzard_ratio_b, aes(x=range_xb, y=h_b)) +
geom_line(color='red', lwd=0.5) +  ggtitle("harzard ratio of Arm_B") + xlab("The number of months") +
ylab("Monthly deaths") + theme_bw()
```



So we can combine above two graphs.

```
ggplot() + geom_line(data = harzard_ratio_a, aes(x=range_xa, y=h_a), color = "black", lwd = 0.5) +
  theme(legend.position = c(0.9,0.7)) +
  geom_line(data = harzard_ratio_b, aes(x = range_xb, y = h_b), color = 'red', lwd = 0.5) +
  ggtitle("harzard ratio of Arm_A and Arm_B") +
  xlab("The number of months") + ylab("Monthly deaths") +
  theme_bw()
```



Problem 9.5

Why does the hypergeometric distribution enter into formula (9.24)?

Solution

In the book, the notations mean n_A = the number of at risk in Arm_A, n_d = the number of deaths, n_B = the number of at risk in Arm_B, n_s = the number of survived patients and y = the number of Arm_A deaths. The data of 2 by 2 display of month-6 for NCOG study are observed after the patients died which means we know n_d. Also, we know the number of patient using the treatments Arm_A and Arm_B respectively which means we know n_A and n_B. Therefore, the sum of each columns and rows 2 by 2 contingency table is fixed. So y determines the other three table entires by substraction, so we are not losing any information by focusing on y. This is why the distribution of y follows the hypergeometric distribution.