

06-23-2023

Hengde Ouyang

2023-06-23

C-statistics for Competing Risk

Previously we have Harrell C & Uno C, the only difference is the weight matrix:

$$\text{Harrell } C : w_{i,j}(G, \tau_G) = \frac{\delta_i I(Y_i < Y_j, Y_i < \tau_G)}{\sum_{i=1}^n \sum_{j=1}^n \delta_i I(Y_i < Y_j, Y_i < \tau_G)}$$

$$\text{Uno } C : w_{i,j}(G, \tau_G) = \frac{\delta_i G\{Y_i\}^{-2} I(Y_i < Y_j, Y_i < \tau_G)}{\sum_{i=1}^n \sum_{j=1}^n \delta_i G\{Y_i\}^{-2} I(Y_i < Y_j, Y_i < \tau_G)}$$

For the Competing Risk, the paper “Concordance for prognostic models with competing risks (2014)” propose two methods :

The first one is a biased estimate:

$$w_{i,j}(G, \tau_G) = \frac{\{I(Y_i < Y_j) + I(Y_i \geq Y_j, \delta_i = 2)\} I(Y_i < \tau_G, \delta_i = 1)}{\sum_{i=1}^n \sum_{j=1}^n \{I(Y_i < Y_j) + I(Y_i \geq Y_j, \delta_i = 2)\}}$$

The second one is inverse probability of censoring weighted (IPCW) estimate:

$$w_{i,j}(G, \tau_G) = \frac{\left\{ \frac{I(Y_i < Y_j)}{\hat{G}(Y_i|X_i)\hat{G}(Y_i|X_j)} + \frac{I(Y_i \geq Y_j, \delta_i = 2)}{\hat{G}(Y_i|X_i)\hat{G}(Y_j|X_j)} \right\} I(Y_i < \tau_G, \delta_i = 1)}{\sum_{i=1}^n \sum_{j=1}^n \left\{ \frac{I(Y_i < Y_j)}{\hat{G}(Y_i|X_i)\hat{G}(Y_i|X_j)} + \frac{I(Y_i \geq Y_j, \delta_i = 2)}{\hat{G}(Y_i|X_i)\hat{G}(Y_j|X_j)} \right\} I(Y_i < \tau_G, \delta_i = 1)}$$

$$\hat{G}(t|X_i) = \exp\left\{-\int_0^t \exp(\hat{\gamma}^T X_i) \hat{\Gamma}_0(s) ds\right\}$$

$\hat{\Gamma}_0$ is the Breslow estimator of the baseline hazard function and $\hat{\gamma}$ is the maximum partial likelihood of the regression coefficients.

As an alternative, we could assume that the censoring is independent of the competing risks outcome and the predictors. And we can use KM estimate for the censoring distribution.

```

RiskC_Wmat1 <- function(Y,delta,tau){
  n <- length(Y)
  Wmat <- matrix(0, n, n)
  for(i in 1:n) {
    if(delta[i] != 0) {
      Wmat[i,] <-
        (delta[i]==1)*
        ((Y[i] < Y)+(Y[i]>= Y & delta == 2))*
        (Y[i] < tau)
    }
  }
  Wmat <- Wmat/sum(Wmat)
  return(Wmat)
}

```

Estimating censoring distribution using KM

```

RiskC_Wmat2 <- function(Y,delta,tau,Cox_G=NULL){

  if (is.null(Cox_G)){
    # An index indicates whether the observation is censored
    censor = ifelse(delta==0,1,0)
    # Censoring Distribution Estimate using Kaplan-Meier Estimator
    KM_G = survfit(formula = Surv(Y ,censor) ~ 1)
    G_y = KM_G$surv[match(Y,KM_G$time)]
  }else{
    # Censoring Distribution Estimate using Kaplan-Meier Estimator
    G_y = Cox_G$surv[match(Y,Cox_G$time)]
  }

  # Get G(Y) for each observation
  # (Since G(Y) in KM_G is ordered we want each G(Y) to match original Y)

  n <- length(Y)
  Wmat <- matrix(0, n, n)
  for(i in 1:n) {
    if(delta[i] != 0) {
      Wmat[i,] <-
        (delta[i]==1)*
        ((Y[i] < Y)*G_y[i]^(-2)+
        (Y[i]>= Y & delta == 2)*G_y[i]^(-1)*
        ifelse(G_y!=0,G_y^(-1),G_y[max(which(G_y != 0))]^(-1)))*
        (Y[i] < tau)
    }
  }
  Wmat <- Wmat/sum(Wmat)
  return(Wmat)
}

```

Test the Code

Reference:

https://www.emilyzabor.com/tutorials/survival_analysis_in_r_tutorial.html#Part_3:_Competing_Risks

```
data(Melanoma, package = "MASS")
```

```
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(gtsummary)
```

```
## #StandWithUkraine
```

```
Melanoma <-
  Melanoma %>%
  mutate(
    status = as.factor(recode(status, `2` = 0, `1` = 1, `3` = 2))
  )
```

status:

0=alive, 1=died from melanoma, 2=dead from other causes.

```
Y = Melanoma$time
delta = Melanoma$status
tau = 4000
A <- model.matrix(time ~ -1+ age + sex + thickness + ulcer,
                  data=Melanoma)

m = 44000
B = 4000
eta = length(Y)
kappa = 1
var.prop = kappa*solve(t(A)%*%A)

beta0 = rep(0,dim(A)[2])
sigma0 = rep(1,dim(A)[2])
```

```

system.time({
result1 = MH_CR_Sampling(Y,delta,tau,
                        A,beta0,sigma0,var.prop,
                        m,B,eta,
                        Wmat_option=1)

result2 = MH_CR_Sampling(Y,delta,tau,
                        A,beta0,sigma0,var.prop,
                        m,B,eta,
                        Wmat_option=2)

censor = ifelse(delta==0,1,0)
Cox_G = survfit(coxph(Surv(Y ,censor) ~ age + sex + thickness + ulcer,
                    data=Melanoma))

result3 = MH_CR_Sampling(Y,delta,tau,
                        A,beta0,sigma0,var.prop,
                        m,B,eta,
                        Wmat_option=2,Cox_G)

})

```

```

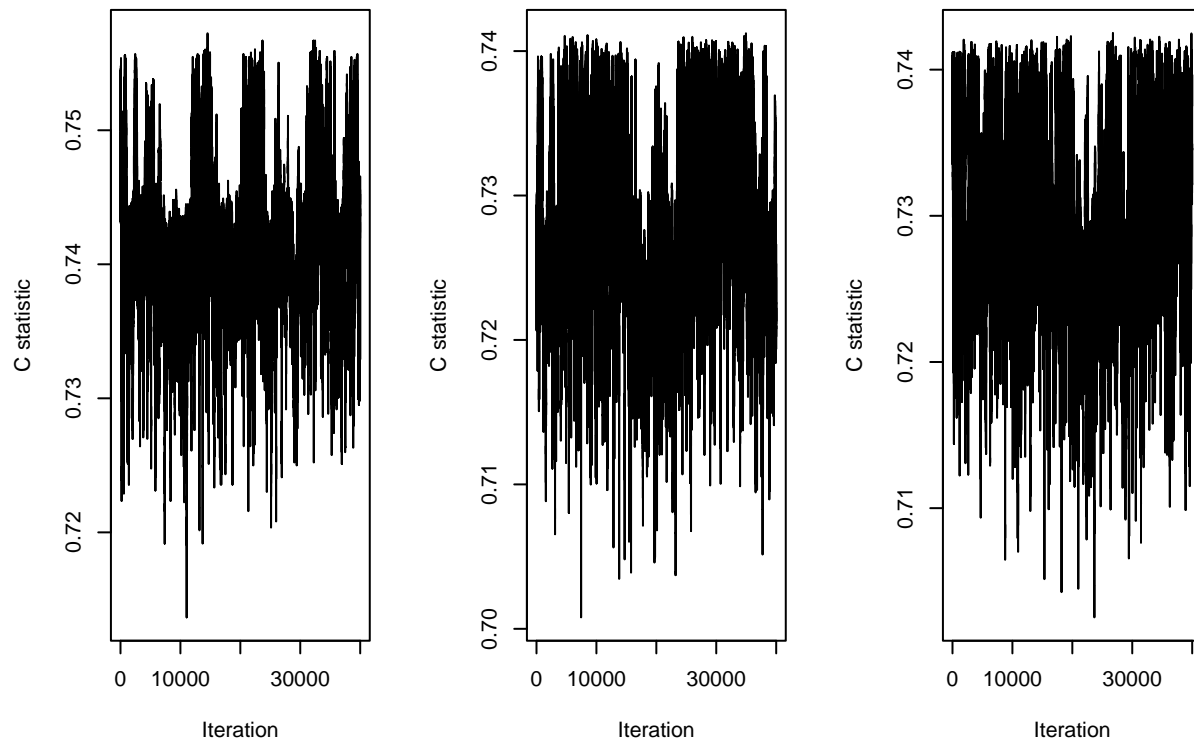
##      user  system elapsed
## 731.37   39.31 1168.06

```

```

par(mfrow=c(1,3))
plot(1:(m-B),result1$C_stat,type="l",xlab="Iteration",ylab="C statistic")
plot(1:(m-B),result2$C_stat,type="l",xlab="Iteration",ylab="C statistic")
plot(1:(m-B),result3$C_stat,type="l",xlab="Iteration",ylab="C statistic")

```



```
Wmat1 = RiskC_Wmat1(Y,delta,tau)
Wmat2 = RiskC_Wmat2(Y,delta,tau)
Wmat3 = RiskC_Wmat2(Y,delta,tau,Cox_G)
C_index(colMeans(result1$THETA),Wmat1)
```

```
## [1] 0.7451304
```

```
C_index(colMeans(result2$THETA),Wmat2)
```

```
## [1] 0.7324692
```

```
C_index(colMeans(result3$THETA),Wmat3)
```

```
## [1] 0.7342648
```

C-statistics for time-dependent covariates

No need to consider?

My thinking: weight matrix for C-statistics only depends on three inputs: the follow-up time,event indicator and a truncation point (Y,delta,tau), covariates only affect the calculation of risk score.

Thus, C statistics can be directly applied to time-dependent covariates.

we only need to think about how to recode the time-dependent covariates or other methods.