Cox-Simulation(08.10)

Hengde Ouyang

2023-08-08

We know the formula:

$$S(t) = exp(-\int_0^t h(u)\ du)$$

Then:

$$S(t|\mathbf{x}_i) = exp(-\int_0^t h(u|\mathbf{x}_i)\ du)$$

In our case:

$$b = exp(\mathbf{x}_i^T\beta),\ h(t|\mathbf{x}_i) = bt^{-0.5},\ S(t|\mathbf{x}_i) = exp(-2b\sqrt{t})$$

Then:

$$F(t|\mathbf{x}_i) = 1 - exp(-2b\sqrt{t})$$

We can generate:

$$T_i = (\frac{-log(1-U_i)}{2b})^2$$

```
Cox_Simulation <- function(n,a,beta,lower,upper,
                           lower_cens, upper_cens,x6=FALSE){
  if (x6 == TRUE){
    x = runif((length(beta)-1)*n,lower,upper)
    x6 = exp(1+rnorm(n))
    X_6 = matrix(x,n,(length(beta)-1))
    X = as.matrix(cbind(X_6[,1:5],x6,X_6[,6:(length(beta)-1)]))

  }else{
```

```
    x = runif(length(beta)*n,lower,upper)
    X = matrix(x,n,length(beta))
  }

  colnames(X) = paste0(1:length(beta))
  # the exponential part
  b = as.numeric(exp(X%*%beta))
  U = runif(n,0,1)
  time = (-0.5*log(1-U)/b)^2

  cens_time <- runif(n, min=lower_cens, max=upper_cens)
  Y <- pmin(time, cens_time)
  delta <- ifelse(time <= cens_time, 1, 0)
  return(list(Y=Y, delta=delta, Time = time,
              X = X))
}

tst <- Cox_Simulation(500, beta=c(-0.2,-0.1,-0.4,3.5,0.4,1.2,0,0,0,0),
                      lower=0, upper=1,
                     lower_cens=0, upper_cens=0.005,x6 = TRUE)
## use tst$Y and tst$delta as the survival outcomes
head(tst$X)
```

```
##              1         2          3         4         5         6          7
## [1,] 0.7112832 0.8901804 0.55847062 0.8205987 0.7117059 0.6538468 0.86939063
## [2,] 0.1847574 0.6141944 0.63050520 0.3011711 0.8452876 1.9239636 0.13020131
## [3,] 0.3878922 0.6799390 0.86197126 0.7862102 0.2646250 3.5651007 0.04215588
## [4,] 0.6877932 0.9893398 0.07359325 0.4524916 0.3182088 2.4145301 0.30072767
## [5,] 0.7844815 0.4467477 0.44020810 0.5619435 0.6462775 2.6182386 0.85631267
## [6,] 0.2322687 0.3537216 0.82467141 0.3040287 0.2452487 8.3370678 0.48063825
##               8          9         10
## [1,] 0.30338273 0.05834356 0.38469513
## [2,] 0.62035743 0.90931978 0.83560346
## [3,] 0.02152032 0.27826776 0.53812627
## [4,] 0.51656370 0.08528936 0.09066494
## [5,] 0.20525126 0.89072954 0.87933991
## [6,] 0.02414409 0.12493155 0.27200635
```

```
library(survival)
Cox = summary(coxph(Surv(tst$Y,tst$delta)~tst$X))
Cox
```

```
## Call:
## coxph(formula = Surv(tst$Y, tst$delta) ~ tst$X)
##
##   n= 500, number of events= 440
##
##              coef exp(coef)  se(coef)      z Pr(>|z|)
## tst$X1  -0.041133  0.959702  0.170768 -0.241  0.80966
## tst$X2   0.039605  1.040400  0.162765  0.243  0.80775
## tst$X3  -0.077934  0.925026  0.166402 -0.468  0.63954
## tst$X4   1.191084  3.290645  0.176742  6.739 1.59e-11 ***
## tst$X5   0.073170  1.075913  0.182604  0.401  0.68864
```

```
## tst$X6   0.137621  1.147541  0.007898 17.425  < 2e-16 ***
## tst$X7   0.429736  1.536852  0.163638  2.626  0.00864 **
## tst$X8   0.364776  1.440191  0.168302  2.167  0.03020 *
## tst$X9   0.097373  1.102271  0.169641  0.574  0.56597
## tst$X10  0.234899  1.264781  0.167793  1.400  0.16153
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##         exp(coef) exp(-coef) lower .95 upper .95
## tst$X1     0.9597     1.0420    0.6867     1.341
## tst$X2     1.0404     0.9612    0.7562     1.431
## tst$X3     0.9250     1.0811    0.6676     1.282
## tst$X4     3.2906     0.3039    2.3272     4.653
## tst$X5     1.0759     0.9294    0.7522     1.539
## tst$X6     1.1475     0.8714    1.1299     1.165
## tst$X7     1.5369     0.6507    1.1152     2.118
## tst$X8     1.4402     0.6944    1.0355     2.003
## tst$X9     1.1023     0.9072    0.7905     1.537
## tst$X10    1.2648     0.7907    0.9103     1.757
##
## Concordance= 0.822  (se = 0.011 )
## Likelihood ratio test= 252.2  on 10 df,   p=<2e-16
## Wald test            = 393.9  on 10 df,   p=<2e-16
## Score (logrank) test = 514.9  on 10 df,   p=<2e-16
```

Prepare the parameter

```
system.time({

  result1 = MH_Sampling(tti,Y,Y.test,delta,delta.test,tau,
                        A,A.test,beta0,sigma0,var.prop,
                        m,B,eta,
                        Wmat_option=0)

})
```

```
##    user  system elapsed
##  101.64   28.28  148.20
```

```
system.time({

  result2 = MH_GP_Sampling(tti,Y,Y.test,delta,delta.test,tau,
                        A,A.all,beta0,alpha0,v0,kappa,
                        m,B,eta,K.all,n,
                        Wmat_option=0)



})
```

```
##    user  system elapsed
```

```
## 4099.87  128.23 5918.47
```

```
system.time({
  result3 = MH_horseshoe_Sampling(tti,Y,Y.test,delta,delta.test,tau,
                                  A,A.test,beta0,sigma0,var.prop,
                                  m,B,eta,v,
                                  Wmat_option=0)

})
```

```
##    user  system elapsed
##   93.23   26.48  164.55
```

```
Wmat = HarrellC_Wmat(Y,delta,tau)
Wmat.test = HarrellC_Wmat(Y.test,delta.test,tau)
```

```
all = data.frame(Model = c("Cox","Linear Regression","Gaussian Process",
                           "LR with horseshoe"),
        C_train = c(C_index(THETA(A,Cox$coefficients[,1]),Wmat),
                    C_index(colMeans(result1$THETA),Wmat),
                    C_index(colMeans(result2$BETA),Wmat),
                    C_index(colMeans(result3$THETA),Wmat)),
        C_test = c(C_index(THETA(A.test,Cox$coefficients[,1]),Wmat.test),
                   C_index(colMeans(result1$THETA.test),Wmat.test),
                   C_index(colMeans(result2$BETA_test),Wmat.test),
                   C_index(colMeans(result3$THETA.test),Wmat.test)),
        Spearman = c(cor(THETA(A.test,Cox$coefficients[,1]),
                         THETA(A.test,c(-0.2,-0.1,-0.4,3.5,0.4,1.2,0,0,0,0)), method="spearman"),
                     cor(colMeans(result1$THETA.test),
                         THETA(A.test,c(-0.2,-0.1,-0.4,3.5,0.4,1.2,0,0,0,0)), method="spearman"),
                     cor(colMeans(result2$BETA_test),
                         THETA(A.test,c(-0.2,-0.1,-0.4,3.5,0.4,1.2,0,0,0,0)), method="spearman"),
                     cor(colMeans(result3$THETA.test),
                         THETA(A.test,c(-0.2,-0.1,-0.4,3.5,0.4,1.2,0,0,0,0)), method="spearman")),
        Kendall = c(cor(THETA(A.test,Cox$coefficients[,1]),
                         THETA(A.test,c(-0.2,-0.1,-0.4,3.5,0.4,1.2,0,0,0,0)), method="kendall"),
                     cor(colMeans(result1$THETA.test),
                         THETA(A.test,c(-0.2,-0.1,-0.4,3.5,0.4,1.2,0,0,0,0)), method="kendall"),
                     cor(colMeans(result2$BETA_test),
                         THETA(A.test,c(-0.2,-0.1,-0.4,3.5,0.4,1.2,0,0,0,0)), method="kendall"),
                    cor(colMeans(result3$THETA.test),
                         THETA(A.test,c(-0.2,-0.1,-0.4,3.5,0.4,1.2,0,0,0,0)), method="kendall")
                     ))
all
```

```
##                Model   C_train    C_test  Spearman   Kendall
## 1                Cox 0.8127224 0.8287657 0.8975338 0.7305051
## 2 Linear Regression 0.8963010 0.9037709 0.9934113 0.9450505
## 3  Gaussian Process 0.8806131 0.8718319 0.9624362 0.8339394
## 4 LR with horseshoe 0.8954144 0.9052133 0.9906391 0.9296970
```

```
parametric_beta = data.frame(Cox = Cox$coefficients[,1],
                             LR = colMeans(result1$BETA),
                             Horse = colMeans(result3$BETA),
                             row.names = paste0("x", 1:10))
parametric_beta
```

```
##             Cox           LR        Horse
## x1  -0.04113293 -0.097272626 -0.022440666
## x2   0.03960537 -0.054971461 -0.014852518
## x3  -0.07793381 -0.198647068 -0.063621188
## x4   1.19108357  0.486784256  0.165772492
## x5   0.07317000  0.005521784 -0.006751818
## x6   0.13762143  0.203567505  0.073745795
## x7   0.42973595 -0.039060984 -0.005035001
## x8   0.36477598  0.022550796  0.005642107
## x9   0.09737289 -0.022440627 -0.006228863
## x10  0.23489878  0.037668183  0.012156127
```