Cox Simulation(08.20)

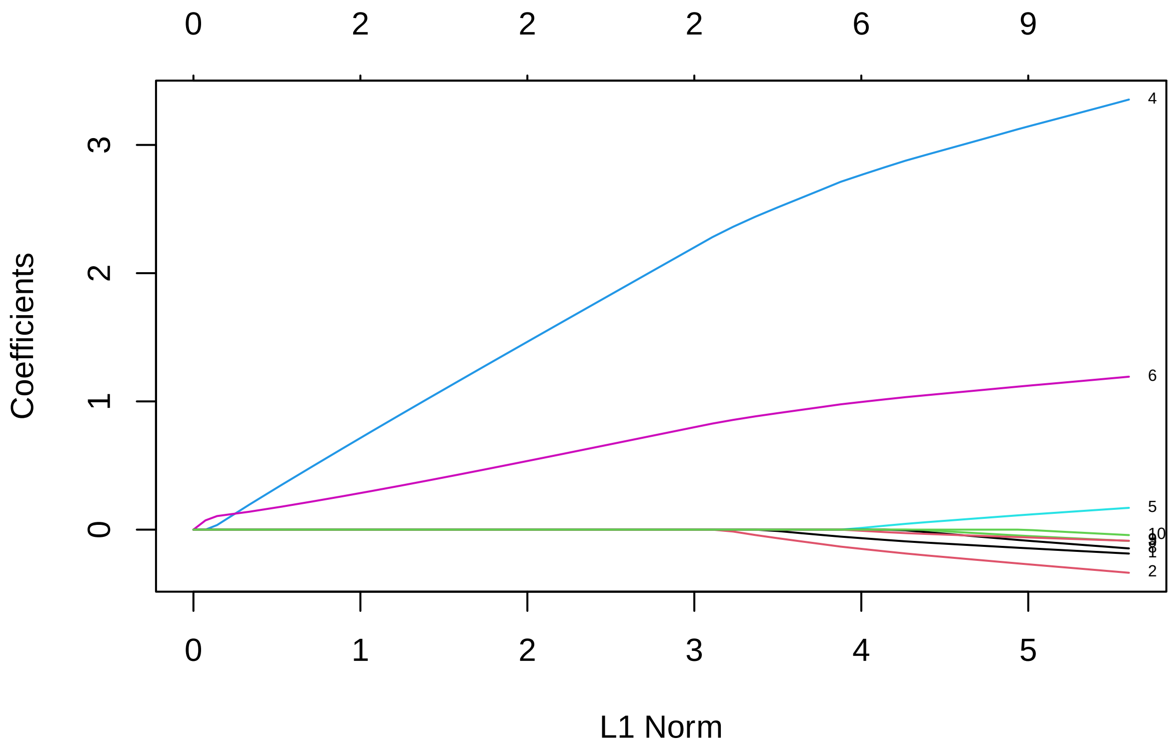
Hengde Ouyang

2023-08-20

```
## Loading required package: Matrix

## Loaded glmnet 4.1-7

##
##  randomForestSRC 3.2.2
##
##  Type rfsrc.news() to see new features, changes, and bug fixes.
##
```

```
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
```

```r
fit <- glmnet(tst$X, Surv(tst$Y,tst$delta),family = "cox")
plot(fit,label = TRUE)
```

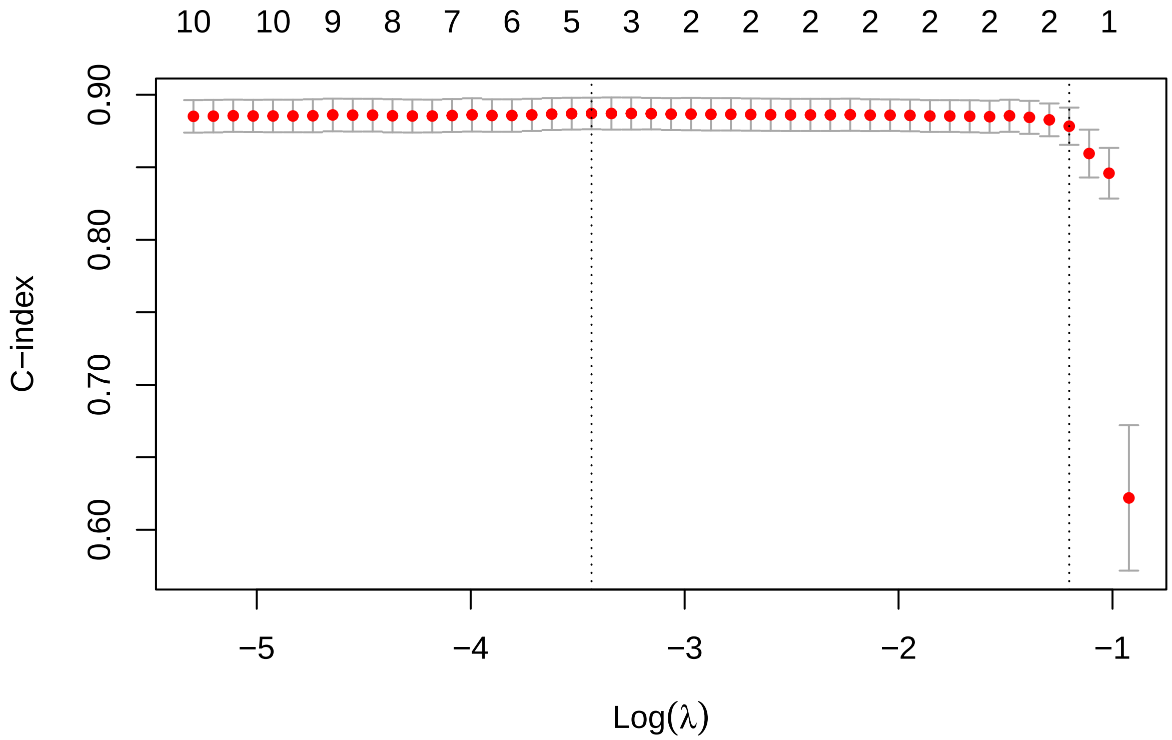


Prepare the data

```r
Cox = summary(coxph(Surv(Y,delta)~scale(A)))
```

```r
fit <- glmnet(A, Surv(Y,delta),family = "cox")
cvfit <- cv.glmnet(A, Surv(Y,delta),family = "cox", type.measure = "C")
```

```r
plot(cvfit)
```



```
B = 500
rsc_data = data.frame(Y=Y,delta=delta,scale(A))
rsc_test_data = data.frame(Y=Y.test,delta=delta.test,scale(A.test))
RF_obj = rfsrc(Surv(Y,delta)~., data=rsc_data,ntree = B, membership = TRUE, importance=TRUE)
RF_obj
```

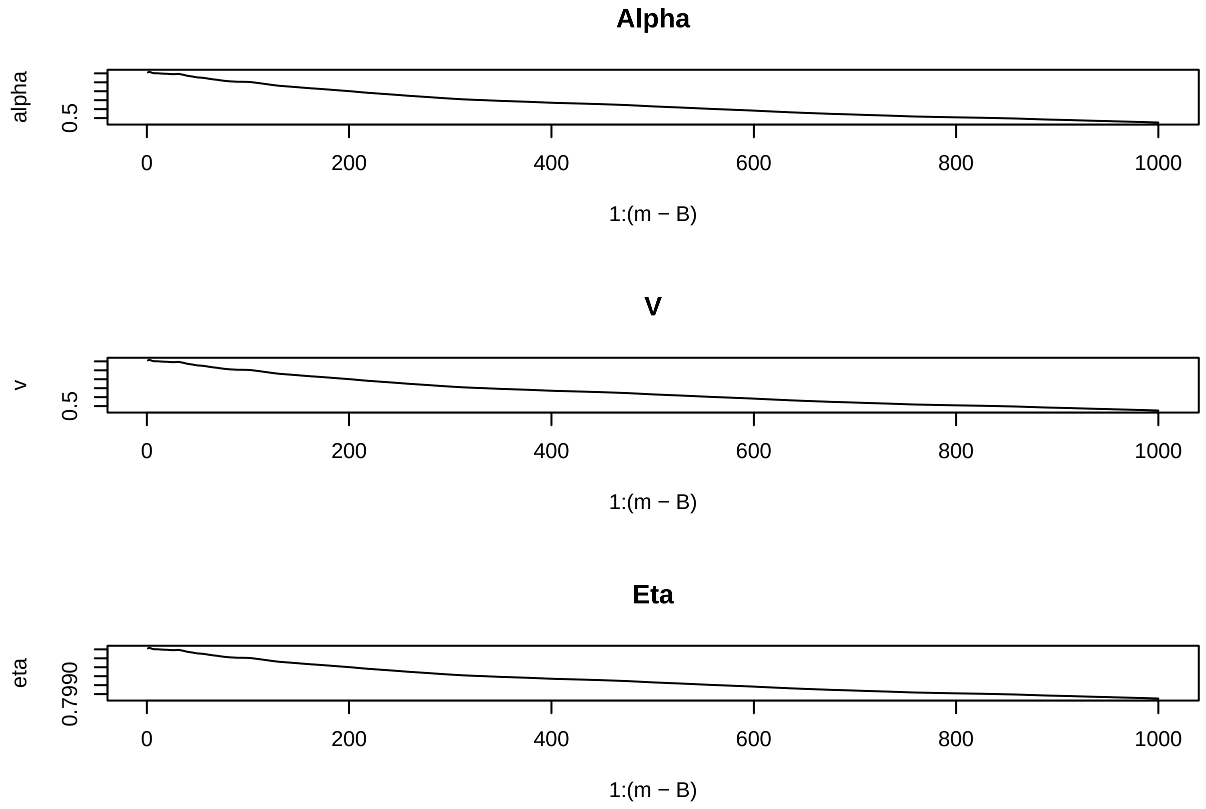
```
##                          Sample size: 400
##                     Number of deaths: 357
##                      Number of trees: 500
##            Forest terminal node size: 15
##        Average no. of terminal nodes: 18.682
## No. of variables tried at each split: 4
##               Total no. of variables: 10
##        Resampling used to grow trees: swor
##     Resample size used to grow trees: 253
##                             Analysis: RSF
##                               Family: surv
##                       Splitting rule: logrank *random*
##        Number of random split points: 10
##                           (OOB) CRPS: 0.06951817
##    (OOB) Requested performance error: 0.14401806
```

```
RF_pred = predict(RF_obj,rsc_test_data)
```

```
system.time({

  m = 1000
  B = 0

  result_learning = Hyper_Learning(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
## 2281.62   95.95 3291.58
```

```
par(mfrow=c(3,1))
plot(1:(m-B),result_learning$ALPHA[,1],type = "l",ylab="alpha",main = "Alpha")
plot(1:(m-B),result_learning$V[,1],type = "l",ylab="v",main = "V")
plot(1:(m-B),result_learning$ETA[,1],type = "l",ylab="eta",main = "Eta")
```



```
system.time({
  m = 11000
  B = 1000
  alpha0 = tail(result_learning$ALPHA[,1],1)
  v0 = tail(result_learning$V[,1],1)
  tail(result_learning$ETA[,1],1)
  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 
## 7195.55  260.25 8690.32
```

```
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 
##  190.19   67.22  295.67
```

```
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 
##  206.77   76.51  324.11
```

```
Wmat = HarrellC_Wmat(Y,delta,tau)
Wmat.test = HarrellC_Wmat(Y.test,delta.test,tau)
sd_design_matrix = apply(tst$X,2,sd)
scale_true_value = c(-0.2,-0.1,-0.4,3.5,0.4,1.2,0,0,0,0)*sd_design_matrix
```

```
all = data.frame(Model = c("Cox(coxph)","Linear Regression","GP after learning",
                           "LR with horseshoe","Cox(glmnet,s=0)",
                           "Cox(glmnet,s=optimal)",
                           "RSF with 500 trees"),
    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_index(THETA(A,as.vector(coef(fit,s = 0))),Wmat),
                C_index(THETA(A,as.vector(coef(fit,s = cvfit$lambda.min))),Wmat),
                C_index(RF_obj$predicted,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),
               C_index(THETA(A.test,as.vector(coef(fit,s = 0))),Wmat.test),
               C_index(THETA(A.test,as.vector(coef(fit,s = cvfit$lambda.min))),Wmat.test),
               C_index(RF_pred$predicted,Wmat.test)),
    Spearman = c(cor(THETA(A.test,Cox$coefficients[,1]),
                     THETA(A.test,scale_true_value), method="spearman"),
                 cor(colMeans(result1$THETA.test),
                     THETA(A.test,scale_true_value), method="spearman"),
                 cor(colMeans(result2$BETA_test),
                     THETA(A.test,scale_true_value), method="spearman"),
                 cor(colMeans(result3$THETA.test),
                     THETA(A.test,scale_true_value), method="spearman"),
                 cor(THETA(A.test,as.vector(coef(fit,s = 0))),
                     THETA(A.test,scale_true_value), method="spearman"),
                 cor(THETA(A.test,as.vector(coef(fit,s = cvfit$lambda.min))),
                     THETA(A.test,scale_true_value), method="spearman"),
                 cor(RF_pred$predicted,
                     THETA(A.test,scale_true_value), method="spearman")),
     Kendall = c(cor(THETA(A.test,Cox$coefficients[,1]),
                     THETA(A.test,scale_true_value), method="kendall"),
                 cor(colMeans(result1$THETA.test),
                     THETA(A.test,scale_true_value), method="kendall"),
                 cor(colMeans(result2$BETA_test),
                     THETA(A.test,scale_true_value), method="kendall"),
                 cor(colMeans(result3$THETA.test),
                     THETA(A.test,scale_true_value), method="kendall"),
                 cor(THETA(A.test,as.vector(coef(fit,s = 0))),
                     THETA(A.test,scale_true_value), method="kendall"),
                 cor(THETA(A.test,as.vector(coef(fit,s = cvfit$lambda.min))),
                     THETA(A.test,scale_true_value), method="kendall"),
                 cor(RF_pred$predicted,
                     THETA(A.test,scale_true_value), method="kendall")
                     ))
all
```

```
##                     Model   C_train    C_test  Spearman   Kendall
## 1            Cox(coxph) 0.7248152 0.7590779 0.7392619 0.5660606
## 2     Linear Regression 0.8905782 0.8741330 0.9952235 0.9563636
## 3     GP after learning 0.8737117 0.8062016 0.8787879 0.7042424
## 4     LR with horseshoe 0.8898366 0.8729090 0.9940594 0.9458586
## 5       Cox(glmnet,s=0) 0.8893123 0.8720930 0.9979598 0.9701010
## 6 Cox(glmnet,s=optimal) 0.8881614 0.8686251 0.9973237 0.9644444
## 7     RSF with 500 trees 0.8984425 0.8610771 0.9517072 0.8250505
```