Package ‘mAFT’

August 23, 2018

Type Package

Title Multi-threshold Accelerate Failure Time Model

Version 0.1

Author Yaguang Li [aut, cre],
Baisuo Jin [aut],
Jialiang Li [aut]

Maintainer Yaguang Li <liyg@mail.ustc.edu.cn>

Depends R (>= 3.5.0), grpreg, plus, lars

Suggests MASS,
knitr,
rmarkdown

Description This package developed a two-stage procedure for simultaneously detecting multiple thresholds and achieving model selection in the segmented accelerate failure time (AFT) model. In the first stage, the threshold problem is formulated as a group model selection problem so that a concave 2-norm group selection method can be applied. In the second stage, the thresholds are finalized via a refining method.

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

R topics documented:

TSMCP 1

Index **6**

`TSMCP` *Two stage multiple change points detection for AFT model.*

Description

This function first formulate the threshold problem as a group model selection problem so that a concave 2-norm group selection method can be applied using the `grpreg` in R packages **grpreg**, and then finalized via a refining method.

Usage

```
TSMCP(Y, X, delta, c)
```

Arguments

Y	the censored logarithm of the failure time
X	design matrix without intercept
delta	the censoring indicator
c	ceiling(c*sqrt(length(Y))) is the length of each segments in spliting stage.

Value

Returns an object with

cp	the change points
coef	the estimated coefficients
sigma	the variance of error
residuals	the residuals
Yn	weighted Y by Kaplan-Meier weight
Xn	weighted Xn by Kaplan-Meier weight

Note

Here Y, X and delta need be re-sorted firstly by the thresholding variable

References

Jialiang Li, Baisuo Jin (2018) Multi-threshold Accelerate Failure Time Model. *The Annals of Statistics*, in press.

See Also

grpreg

Examples

```
## example 1,two thresholds.
## generate data
n=100

X=matrix(rnorm(n*5,0,1),n,5)


#Real threshods (qnorm(0.3),qnorm(0.6))=(-0.5244,0.2533)
id1=which(X[,1]<=qnorm(0.3))
id2=which(X[,1]<=qnorm(0.6) & X[,1]>qnorm(0.3))
id3=which(X[,1]>qnorm(0.6))

C=apply(X,1,sum)+rnorm(n,2,4)


beta01=2
```

```
beta11=c(rep(1,5))

beta02=1
beta12=c(1,1,0,rep(0,2))

beta03=1
beta13=c(0,2,0,rep(0,2))
#Real coefficeints:(beta01,beta11,beta02-beta01,beta12-beta11,beta03-beta02,beta13-beta12)
#=(2,1,1,1,1,1,-1,0,0,-1,-1,-1,0,-1,1,0,0,0)

X1=X[id1,]
n1=length(id1)


T1=X1%*%beta11+beta01+rnorm(n1,0,sqrt(0.5))

C1=C[id1]
delta1=C1
Y1=T1
for(i in 1:n1)
{
  if(T1[i]<C1[i])
  {
    delta1[i]=1
  }
  if(T1[i]>=C1[i])
  {
    delta1[i]=0
    Y1[i]=C1[i]
  }
}

Z11=cbind(Y1,X1,delta1,C1)

X2=X[id2,]
n2=length(id2)

T2=X2%*%beta12+beta02+rnorm(n2,0,sqrt(0.5))

C2=C[id2]
delta2=C2
Y2=T2
for(i in 1:n2)
{
  if(T2[i]<C2[i])
  {
    delta2[i]=1
  }
  if(T2[i]>=C2[i])
  {
    delta2[i]=0
    Y2[i]=C2[i]
  }
}

Z12=cbind(Y2,X2,delta2,C2)
```

```
X3=X[id3,]
n3=length(id3)


T3=X3%*%beta13+beta03+rnorm(n3,0,sqrt(0.5))

C3=C[id3]
delta3=C3
Y3=T3
for(i in 1:n3)
{
  if(T3[i]<C3[i])
  {
    delta3[i]=1
  }
  if(T3[i]>=C3[i])
  {
    delta3[i]=0
    Y3[i]=C3[i]
  }
}

Z13=cbind(Y3,X3,delta3,C3)



ZZ=rbind(Z11,Z12,Z13)
## ZZ[,2] is the thresholding variable
ord=order(ZZ[,2])
ZZ=ZZ[ord,]


n=dim(ZZ)[1]
p=dim(ZZ)[2]-3
Y=ZZ[,1]
X=ZZ[,2:(p+1)]
delta=ZZ[,p+2]



n=length(Y)
p=dim(X)[2]
id1=which(delta==1)
n1=length(id1)


##estimate thresholds

c=seq(0.5,1.5,0.1)
m=ceiling(c*sqrt(n1))
c=c[which(m>p+1)]


bicy=c
tsmc=NULL
```

```
for(i in 1:length(c))
{

  tsm=TSMCP(Y,X,delta,c[i])
  bicy[i]=log(n)*((length(tsm[[1]])+1)*(p+1))+n*log(tsm[[3]])

  tsmc[[i]]=tsm
}

tsmcp=tsmc[[which(bicy==min(bicy))[1]]]
#choose the optimal results by BIC
tsmcp[[1]] # change points
X[tsmcp[[1]],1] #thresholds. Real threshods (qnorm(0.3),qnorm(0.6))=(-0.5244,0.2533)
tsmcp[[2]] # coefficients. Real value:(2,1,1,1,1,1,-1,0,0,-1,-1,-1,0,-1,1,0,0,0)
tsmcp[[3]] #variance of error. real variance of error is 0.5
min(bicy)  # bic
```

Index

grpreg, *1*

TSMCP, 1