Package 'NPMLENCC'

May 31, 2019

Title Non-Parametric Maximum Likelihood Estimate for Cohort Samplings

Version 1.0

Description To compute the non-parametric maximum likelihood estimates (NPMLEs) and penalized NPMLEs with SCAD, HARD and LASSO penalties for nested case-control or case-cohort sampling design with time matching under Cox's regression model. It also proposes the standard error formula for estimator using observed profile likelihood. For details about (penalized) NPNLEs see the original paper ``Penalized Full Likelihood Approach to Variable Selection for Cox's Regression Model under Nested Case-Control Sampling" by Wang et al. (2019) <doi:10.1007/s10985-019-09475-z>.

Author Jie-Huei Wang, Chun-Hao Pan, I-Shou Chang, and Chao A. Hsiung

Maintainer Jie-Huei Wang <jhwang@stat.sinica.edu.tw>

Date 2019-05-27

Depends R (>= 3.4.3), MASS, survival, splines

License GPL (>= 2)

Encoding UTF-8

LazyData true

Repository CRAN

RoxygenNote 6.1.1

NeedsCompilation no

Date/Publication 2019-05-31 08:30:03 UTC

R topics documented:

TNPMLE *Non-Parametric Maximum-Likelihood Estimation for Cohort Samplings with Time Matching under Cox's Regression Model*

Description

The function utilizes a self-consistency iterative algorithm to calculate NPMLEs for cohort samplings with time matching under Cox's regression model. In addition to compute NPMLEs, it can also estimate asymptotic varance, as described in Wang et al. (2019). The Cox's regression model is

$$\lambda(t|z) = \lambda_0(t)\exp(z^T\beta).$$

Usage

```
TNPMLE(data, iteration1, iteration2, converge)
```

Arguments

`data`	The $N \times P$ matrix of data. There are N individuals in matrix, with one individual in each row. The P columns orderly included the observable times which are time-to-event or censoring times and without ties at the event times, the status is a binary variable with 1 indicating the event has occured and 0 indicating (right) censoring, and the $(P-2)$ covariates which only observed for some individuals. Note that the covariate of those unobserved individuals are denoted by -9, not missing value (NA) and the observed covariates values are not the same as -9.
`iteration1`	The number of iteration for computing NPMLEs.
`iteration2`	The number of iteration for computing profile likelihoods which are used to estimate asymptotic variance.
`converge`	The parameter influence the convergence of the algorithm, if the sup-norm of $\left(\hat{\beta}_{(k)} - \hat{\beta}_{(k-1)}\right)$ is smaller than the thresholding value, we then declare the estimates are converge, stop computing estimates, otherwise the number of iteration for computing estimates is the `iteration1` term.

Value

Returns a list with components

`num`	The numbers of case and observed subjects.
`iloop`	The final number of iteration for computing NPMLEs.
`diff`	The sup-norm distance between the last two iterations of the estimates of the relative risk coefficients.
`likelihood`	The log likelihood value of NPMLEs.
`npmle`	The estimated regression coefficients with their corresponding estimated standard errors and p-values.

`Lnpmle`	The estimated cumulative baseline hazards function.
`Pnpmle`	The empirical distribution of covariates which are missing for unobserved subjects.
`elements`	A list which is used to plot cumulative baseline hazards function and baseline survival function. The $n \times 3$ matrix of data, n is the total number of case and the 3 columns orderly included the order observed time of case, the estimated cumulative baseline hazards function and estimated baseline survival function.
`Adata`	Arranging original data to let our analysis performed conveniently. There are three steps for this arrangement, the 1st step divides original data into observed and unobserved groups, then put them on top and bottom, respectively; the 2nd step divides the observed data of 1st step into case and control groups; the final step order the case data of 2nd step by observed time from low to high.

Note

The missing value (NA) in the DATA is not allowed in this version.

References

Wang JH, Pan CH, Chang IS*, and Hsiung CA (2019) Penalized full likelihood approach to variable selection for Cox's regression model under nested case-control sampling. published in Lifetime Data Analysis <doi:10.1007/s10985-019-09475-z>.

See Also

See TPNPMLE.

Examples

```
set.seed(100)
library(splines)
library(survival)
library(MASS)
beta=c(1,0)
lambda=0.3
cohort=100
covariate=2+length(beta)
z=matrix(rnorm(cohort*length(beta)),nrow=cohort)
rate=1/(runif(cohort,1,3)*exp(z%*%beta))
c=rexp(cohort,rate)
u=-log(runif(cohort,0,1))/(lambda*exp(z%*%beta))
time=apply(cbind(u,c),1,min)
status=(u<=c)+0
casenum=sum(status)
odata=cbind(time,status,z)
odata=data.frame(odata)
a=order(status)
data=matrix(0,cohort,covariate)
data=data.frame(data)
for (i in 1:cohort){
```

```
data[i,]=odata[a[cohort-i+1],]
}
ncc=matrix(0,cohort,covariate)
ncc=data.frame(data)
aa=order(data[1:casenum,1])
for (i in 1:casenum){
ncc[i,]=data[aa[i],]
}
control=1
q=matrix(0,casenum,control)
for (i in 1:casenum){
k=c(1:cohort)
k=k[-(1:i)]
sumsc=sum(ncc[i,1]<ncc[,1][(i+1):cohort])
if (sumsc==0) {
q[i,]=c(1)
} else {
q[i,]=sample(k[ncc[i,1]<ncc[,1][(i+1):cohort]],control)
}
}
cacon=c(q,1:casenum)
k=c(1:cohort)
owf=k[-cacon]
wt=k[-owf]
owt=k[-wt]
ncct=matrix(0,cohort,covariate)
ncct=data.frame(ncct)
for (i in 1:length(wt)){
ncct[i,]=ncc[wt[i],]
}
for (i in 1:length(owt)){
ncct[length(wt)+i,]=ncc[owt[i],]
}
d=length(wt)+1
ncct[d:cohort,3:covariate]=-9
TNPMLEtest=TNPMLE(data=ncct,iteration1=100,iteration2=30,converge=0)
```

TPNPMLE *Penalized Non-Parametric Maximum-Likelihood Estimation (PNPMLEs) for Cohort Samplings with Time Matching under Cox's Regression Model*

Description

The function utilizes a self-consistency iterative algorithm to calculate PNPMLEs by adding penalty function for cohort samplings with time matching under Cox's regression model. In addition to compute PNPMLEs, it can also estimate asymptotic varance, as described in Wang et al. (2019+). The Cox's regression model is

$$\lambda(t|z) = \lambda_0(t)\exp(z^T\beta).$$

Usage

```
TPNPMLE(data, iteration1, iteration2, converge, penalty, penaltytuning,
  fold, cut, seed)
```

Arguments

data	The description is the same as the statement of TNPMLE function.
iteration1	The number of iteration for computing (P)NPMLEs.
iteration2	The number of iteration for computing profile likelihoods which are used to estimate asymptotic variance.
converge	The description is the same as the statement of TNPMLE function.
penalty	The choice of penalty, it can be SCAD, HARD or LASSO.
penaltytuning	The tuning parameter for penalty function, it is a sequence of numeric vector.
fold	The fold information for cross validation. Without loss of generality, we note that fold value have to be bigger than one (>1) and cohort size is divisible by fold value. However, if cohort size is not able to be divided, we are going to partition off cohort into several suitable parts according to fold value automaticly for cross-validation.
cut	The cut point. When $\hat{\beta}_j$ is smaller than the cut point, we set $\hat{\beta}_j$ be zero, i.e. remove the corresponding covariate from our model to do variable selection.
seed	The seed of the random number generator to obtain reproducible results.

Value

Returns a list with components

num	The numbers of case and observed subjects.
iloop	The final number of iteration for computing PNPMLEs.
diff	The sup-norm distance between the last two iterations of the estimates of the relative risk coefficients.
cvl	The cross-validated profile log-likelihood.
tuning	The suitable tuning parameter, such that the maximum of cross-validated profile log-likelihood is attained.
likelihood	The log likelihood value of PNPMLEs.
pnpmle	The estimated regression coefficients with their corresponding estimated standard errors and p-values.
Lpnpmle	The estimated cumulative baseline hazards function.
Ppnpmle	The empirical distribution of covariates which are missing for unobserved subjects.
elements	The description is the same as the statement of TNPMLE function.
Adata	The description is the same as the statement of TNPMLE function.

Note

The missing value (NA) in the DATA is not allowed in this version.

References

Wang JH, Pan CH, Chang IS*, and Hsiung CA (2019) Penalized full likelihood approach to variable selection for Cox's regression model under nested case-control sampling. published in Lifetime Data Analysis <doi:10.1007/s10985-019-09475-z>.

See Also

See TNPMLE.

Examples

```
set.seed(100)
library(splines)
library(survival)
library(MASS)
beta=c(1,0)
lambda=0.3
cohort=100
covariate=2+length(beta)
z=matrix(rnorm(cohort*length(beta)),nrow=cohort)
rate=1/(runif(cohort,1,3)*exp(z%*%beta))
c=rexp(cohort,rate)
u=-log(runif(cohort,0,1))/(lambda*exp(z%*%beta))
time=apply(cbind(u,c),1,min)
status=(u<=c)+0
casenum=sum(status)
odata=cbind(time,status,z)
odata=data.frame(odata)
a=order(status)
data=matrix(0,cohort,covariate)
data=data.frame(data)
for (i in 1:cohort){
data[i,]=odata[a[cohort-i+1],]
}
ncc=matrix(0,cohort,covariate)
ncc=data.frame(data)
aa=order(data[1:casenum,1])
for (i in 1:casenum){
ncc[i,]=data[aa[i],]
}
control=1
q=matrix(0,casenum,control)
for (i in 1:casenum){
k=c(1:cohort)
k=k[-(1:i)]
sumsc=sum(ncc[i,1]<ncc[,1][(i+1):cohort])
if (sumsc==0) {
q[i,]=c(1)
```

```
} else {
q[i,]=sample(k[ncc[i,1]<ncc[,1][(i+1):cohort]],control)
}
}
cacon=c(q,1:casenum)
k=c(1:cohort)
owf=k[-cacon]
wt=k[-owf]
owt=k[-wt]
ncct=matrix(0,cohort,covariate)
ncct=data.frame(ncct)
for (i in 1:length(wt)){
ncct[i,]=ncc[wt[i],]
}
for (i in 1:length(owt)){
ncct[length(wt)+i,]=ncc[owt[i],]
}
d=length(wt)+1
ncct[d:cohort,3:covariate]=-9
TPNPMLEtest=TPNPMLE(ncct,100,30,0,"SCAD",seq(0.10,0.13,0.005),2,1e-05,1)
```

Index