rm(list=ls())

library(MASS)

library(coxme)

library(survival)

library(flexclust)

options(warn=-1)

Sn=function(x,t){  
xx=1-t/sqrt(sum(x^2))

if(xx>0){

s=xx\*x

}else{

s=0\*x

}

list(s=s)

}

Theta\_SCAD=function(u,lambda){

L=length(u)

Ta=rep(0,L)

for(k in 1:L){

uabs=abs(u[k])

if(uabs<=(lambda+lambda/eta)){

Ta[k]=Sn(x=u[k],t=lambda/eta)$s

}else if(uabs>(lambda+lambda/eta)&uabs<=(v\*lambda)){

Ta[k]=Sn(x=u[k],t=v\*lambda/((v-1)\*eta))$s/(1-1/((v-1)\*eta))

}else{

Ta[k]=u[k]

}

}

list(Ta=Ta)

}

Fn\_SCAD=function(lambda){

ahat=a0

betahat=b0

theta=theta0

upsilon=upsilon0

counter=1

eps\_rel=0.0001

eps\_abs=0.0001

repeat{

a1=as.vector(Z%\*%ahat+X%\*%betahat)

u=rep(0,N)

A=matrix(0,N,N)

for(k in 1:N){

f1=sapply(Cset[[k]],function(x) sum(exp(a1[Risk[[x]]])))

u[k]=status[k]-sum(exp(a1[k])/f1)

A[k,k]=sum((exp(a1[k])\*f1-exp(2\*a1[k]))/f1^2)

}

Yt=a1+u/diag(A)

xx=solve(t(X)%\*%A%\*%X)

Qx=A-A%\*%X%\*%xx%\*%t(X)%\*%A

ahat=solve(t(Z)%\*%Qx%\*%Z+eta\*t(Delta)%\*%Delta)%\*%(t(Z)%\*%Qx%\*%Yt+eta\*t(Delta)%\*%(theta-1/eta\*upsilon))

ahat=ahat-mean(ahat)

betahat=solve(t(X)%\*%A%\*%X)%\*%t(X)%\*%A%\*%(Yt-Z%\*%ahat)

theta1=theta

for(i in 1:(m-1)){

Pi=ahat[i]-ahat[(1:m)>i]+1/eta\*upsilon[((i-1)\*(2\*m-i)/2+1):((i-1)\*(2\*m-i)/2+m-i)]

theta[((i-1)\*(2\*m-i)/2+1):((i-1)\*(2\*m-i)/2+m-i)]=Theta\_SCAD(Pi,lambda)$Ta

upsilon[((i-1)\*(2\*m-i)/2+1):((i-1)\*(2\*m-i)/2+m-i)]=upsilon[((i-1)\*(2\*m-i)/2+1):((i-1)\*(2\*m-i)/2+m-i)]+eta\*(ahat[i]-ahat[(1:m)>i]-theta[((i-1)\*(2\*m-i)/2+1):((i-1)\*(2\*m-i)/2+m-i)])

}

r=Delta%\*%ahat-theta

s=eta\*t(Delta)%\*%(theta-theta1)

primal=sqrt(sum(r^2))

dual=sqrt(sum(s^2))

if(primal>dual){

eta=2\*eta

}else{

eta=eta

}

eps\_p=eps\_abs\*sqrt(m\*(m-1)/2)+eps\_rel\* max(sqrt(sum((Delta%\*%ahat)^2)),sqrt(sum((theta)^2)))

eps\_d=eps\_abs\*sqrt(m)+eps\_rel\*sqrt(sum((t(Delta)%\*%upsilon)^2))

if(primal<eps\_p&dual<eps\_d){

break

}

if(counter>200){

break

}

counter=counter+1

print(counter)

}

list(ahat=ahat,betahat=betahat)

}

T=1

nrep=100

m=50

p=2

c1=m\*0.18

c2=m\*0.18

a1=-1

a2=1

eta=1

v=3

ngroup=rep(0,nrep)

aest=matrix(0,nrep,m)

betaest=matrix(0,nrep,p)

aMSE=rep(0,nrep)

se.beta=matrix(0,nrep,p)

cp.beta=matrix(0,nrep,p)

RI=matrix(0,nrep,4)

se.a=NULL

cp.a=NULL

Aest=NULL

Mu=NULL

betaran=matrix(0,nrep,p)

se.beta\_ran=matrix(0,nrep,p)

cp.beta\_ran=matrix(0,nrep,p)

aMSE\_ran=rep(0,nrep)

aran=matrix(0,nrep,m)

betafix=matrix(0,nrep,p)

se.beta\_fix=matrix(0,nrep,p)

cp.beta\_fix=matrix(0,nrep,p)

aMSE\_fix=rep(0,nrep)

afix=matrix(0,nrep,m)

aor=matrix(0,nrep,m)

betaor=matrix(0,nrep,p)

se.beta\_or=matrix(0,nrep,p)

cp.beta\_or=matrix(0,nrep,p)

aMSE\_or=rep(0,nrep)

Aor=matrix(0,nrep,2)

se.a\_or=matrix(0,nrep,2)

cp.a\_or=matrix(0,nrep,2)

SERmse\_fix=rep(0,nrep)

SERmse\_ran=rep(0,nrep)

SERmse=rep(0,nrep)

SERmse\_or=rep(0,nrep)

agrid=NULL

betagrid=NULL

Im=diag(m)

Delta=NULL

for(i in 1:(m-1)){

Delta=cbind(Delta,(Im[,i]-Im[,(1:m)>i]))

}

Delta=t(Delta)

xi=seq(0.2,0.3,by=0.03)

nxi=length(xi)

for(irep in 1:nrep){

set.seed((T-1)\*nrep+irep)

ni=round(runif(m,10,100))

N=sum(ni)

atrue=c(rnorm(c1,-1,0.1),rnorm(m-c1-c2,0,0.1),rnorm(c2,1,0.1))

J1=matrix(rep(1:p,times=p),byrow=FALSE, nrow=p)

K1=matrix(rep(1:p,times=p),byrow=TRUE, nrow=p)

sigma=0.2^abs(J1-K1)

X=mvrnorm(N,rep(0,p),sigma)

beta=c(0.2,-0.2)

Z=matrix(0,N,m)

id=NULL

a=NULL

SERtrue=rep(0,m)

for(k in 1:m){

Z[(sum(ni[(1:m)<k])+1):(sum(ni[(1:m)<k])+ni[k]),k]=rep(1,ni[k])

id=c(id,rep(k,ni[k]))

a=c(a,rep(atrue[k],ni[k]))

SERtrue[k]=sum(exp(atrue[k]+X[(sum(ni[(1:m)<k])+1):(sum(ni[(1:m)<k])+ni[k]),]%\*%beta))/sum(exp(median(atrue)+X[(sum(ni[(1:m)<k])+1):(sum(ni[(1:m)<k])+ni[k]),]%\*%beta))

}

u=runif(N,0,1)

scale=0.05

survtime=as.vector(-log(u)/(scale\*exp(Z%\*%atrue+X%\*%beta)))

censor=runif(N,0,50)

time=pmin(survtime,censor)

status=ifelse(time<censor,1,0)

rate=1-sum(status)/length(status)

Data=data.frame(cbind(id,X,a,survtime,status,time))

Risk=list()

Cset=list()

for(i in 1:N){

Risk[[i]]=which(time>=time[i])

Cset[[i]]=which(time<=time[i])

}

SER=function(x,ahat,betahat){

SER=sum(exp(ahat[x]+X[(sum(ni[(1:m)<x])+1):(sum(ni[(1:m)<x])+ni[x]),]%\*%betahat))/sum(exp(median(ahat)+X[(sum(ni[(1:m)<x])+1):(sum(ni[(1:m)<x])+ni[x]),]%\*%betahat))

}

#random-effect model

model=coxme(Surv(time,status)~X+(1|id),data=Data)

betaran[irep,]=fixef(model)

aran[irep,]=as.vector(ranef(model)$id)

aMSE\_ran[irep]=sum((atrue-aran[irep,])^2)/m

se.beta\_ran[irep,]=sqrt(diag(vcov(model)))

cp.beta\_ran[irep,]=ifelse(abs(betaran[irep,]-beta)<1.96\*se.beta\_ran[irep,],1,0)

SERran=sapply(1:m,SER,ahat=aran[irep,],betahat=betaran[irep,])

SERmse\_ran[irep]=sum((SERtrue-SERran)^2)/m

#fixed-effect model

r1=m/2

fix=coxph(Surv(time,status)~X+relevel(factor(id),ref=r1),data=Data)

betafix[irep,]=coef(fix)[1:p]

se.beta\_fix[irep,]=summary(fix)$coef[1:p,3]

cp.beta\_fix[irep,]=ifelse(abs(betafix[irep,]-beta)<1.96\*se.beta\_fix[irep,],1,0)

afix[irep,]=as.vector(c(coef(fix)[-(1:p)][1:(r1-1)],0,coef(fix)[-(1:p)][r1:(m-1)]))

aMSE\_fix[irep]=sum((atrue-afix[irep,])^2)/m

SERfix=sapply(1:m,SER,ahat=afix[irep,],betahat=betafix[irep,])

SERmse\_fix[irep]=sum((SERtrue-SERfix)^2)/m

#Oracle

gk=list()

id\_or=rep(0,N)

gk[[1]]=1:c1

gk[[2]]=(c1+1):(m-c2)

gk[[3]]=(m-c2+1):m

id\_or[unlist(lapply(gk[[1]],function(t)which(id==t)))]=1

id\_or[unlist(lapply(gk[[2]],function(t)which(id==t)))]=2

id\_or[unlist(lapply(gk[[3]],function(t)which(id==t)))]=3

data2=data.frame(id\_or,X,a,survtime,status,time)

model\_or=coxph(Surv(time,status)~X+relevel(factor(id\_or),"2"),data=data2)

para\_or=as.numeric(coef(model\_or))

se\_or=as.numeric(summary(model\_or)$coef[,3])

betaor[irep,]=para\_or[1:p]

aor[irep,gk[[1]]]=para\_or[p+1]

aor[irep,gk[[3]]]=para\_or[p+2]

se.beta\_or[irep,]=se\_or[1:p]

cp.beta\_or[irep,]=ifelse(abs(betaor[irep,]-beta)<1.96\*se.beta\_or[irep,],1,0)

aMSE\_or[irep]=sum((atrue-aor[irep,])^2)/m

Aor[irep,]=para\_or[-(1:p)]

se.a\_or[irep,]=se\_or[-(1:p)]

cp.a\_or[irep,]=ifelse(abs(para\_or[-(1:p)]-c(a1,a2))<1.96\*se\_or[-(1:p)],1,0)

SERor=sapply(1:m,SER,ahat=aor[irep,],betahat=betaor[irep,])

SERmse\_or[irep]=sum((SERtrue-SERor)^2)/m

#fused-effects model

a0=as.vector(coef(fix)[-(1:p)])

a0=c(a0[1:(r1-1)],0,a0[r1:(m-1)])

b0=as.matrix(as.numeric(betafix[irep,]))

up0=rep(0,m)

theta0=NULL

upsilon0=NULL

for(i in 1:(m-1)){

theta0=c(theta0,(a0[i]-a0[(1:m)>i]))

upsilon0=c(upsilon0,(up0[i]-up0[(1:m)>i]))

}

Group=rep(0,nxi)

bic=rep(0,nxi)

b1=rep(0,nxi)

b2=rep(0,nxi)

ascad=list()

betascad=list()

for(i in 1:nxi){

value=Fn\_SCAD(lambda=xi[i])

ascad[[i]]=value$ahat

betascad[[i]]=value$betahat

ai=ascad[[i]]

j=1

group=list()

group[[1]]=which(abs(ai-ai[1])<0.2)

while(length(unlist(group))<m){

ai[-unlist(group)][which(abs(ai[-unlist(group)]-ai[-unlist(group)][1])<0.2)]=j

group[[j+1]]=which(ai==j)

j=j+1

}

mu=as.numeric(lapply(1:length(group),function(x)1/length(group[[x]])\*sum(ascad[[i]][group[[x]]])))

df=cbind(lengths(group),mu)

df=df[order(df[,2]),]

print(df)

Group[i]=length(group)

LogL=function(t){

risk=Risk[[t]]

l2=log(sum(exp(Z%\*%ascad[[i]]+X%\*%betascad[[i]])[risk,]))

return(l2)

}

l2=apply(matrix(c(1:N),N,1),1,LogL)

b1[i]=-2\*sum(status\*(Z%\*%ascad[[i]]+X%\*%betascad[[i]]-l2))

Cn=log(log(N+p))

b2[i]=log(N)\*(Group[i]+p)

bic[i]=b1[i]+Cn\*b2[i]

}

I=which.min(bic)

aI=ascad[[I]]

j=1

group0=list()

group0[[1]]=which(abs(aI-aI[1])<0.3)

while(length(unlist(group0))<m){

aI[-unlist(group0)][which(abs(aI[-unlist(group0)]-aI[-unlist(group0)][1])<0.3)]=j

group0[[j+1]]=which(aI==j)

j=j+1

}

mu0=as.numeric(lapply(1:length(group0),function(x)1/length(group0[[x]])\*sum(ascad[[I]][group0[[x]]])))

group=lapply(order(mu0),function(x)group0[[x]])

mu=mu0[order(mu0)]

df=cbind(lengths(group),mu)

print(df)

ngroup[irep]=length(group)

id1=rep(0,N)

Z1=matrix(0,m,ngroup[irep])

for(k in 1:ngroup[irep]){

Z1[group[[k]],k]=1

id1[unlist(lapply(group[[k]],function(x)(sum(ni[1:x])-ni[x]+1):sum(ni[1:x])))]=k

}

r0=which.min(abs(mu))

data1=data.frame(X,id1,time,status)

model1=coxph(Surv(time,status)~X+relevel(factor(id1),ref=r0),data=data1)

summary(model1)

para=as.numeric(coef(model1))

se=as.numeric(summary(model1)$coef[,3])

aest[irep,]=Z1[,-r0]%\*%as.matrix(para[-(1:p)])

betaest[irep,]=para[1:p]

se.beta[irep,]=se[1:p]

cp.beta[irep,]=ifelse(abs(betaest[irep,]-beta)<1.96\*se.beta[irep,],1,0)

aMSE[irep]=sum((atrue-aest[irep,])^2)/m

RI[irep,]=comPart(round(aest[irep,]),round(atrue))

SERest=sapply(1:m,SER,ahat=aest[irep,],betahat=betaest[irep,])

SERmse[irep]=sum((SERtrue-SERest)^2)/m

if(ngroup[irep]==3){

Aest=rbind(Aest,para[-(1:p)])

se.a=rbind(se.a,se[-(1:p)])

cp.a=rbind(cp.a,ifelse(abs(para[-(1:p)]-c(a1,a2))<1.96\*se[-(1:p)],1,0))

Mu=rbind(Mu,mu)

}

}