Here is a brief explanation of the CPADMM package.

**Three main functions:** **CPADMME**(for various elastic-net regressions), **CPADMMF**(for various sparse fused regressions), **CPADMMG**(for various sparse group regressions).

**CPADMM(E or F or G)(X, y, loss, pen, lambda1, lambda2, tau, ka, maxite, M)**

# X：Matrix of predictors, of dimension (n\*p); each row is an observation.

# y: Response variable.

#loss: A character string specifying the loss function to use. Available options are least squares loss (LS), square root loss (SR), Huber loss (Huber) and quantile loss (Quan).

# pen: A character string specifying the first penalty to use. Available options are L1(LASSO), SCAD and MCP.

# lambda1: The tuning parameter for the first penalty.

# lambda2: The tuning parameter for the second penalty.

# tau: The quantile level τ. The value must be in (0,1).

# ka: The constants need to be given in Huber loss.

# maxite: Maximum number of iterations allowed in the ADMM algorithm at fixed lambda values. If the algorithm does not converge, consider increasing maxite.

# M: The number of local machines.

**Two examples for elastic-net regressions**

#######Example 1(High-dimensional regression)

s=1# s=1,2,3,4,5,...

n=720

p=2560\*s

beta=rep(0,p)

for (j in 1:p) {

beta[j] = (-1)^j\*exp(-(2\*j-1)/20)

}

rho <- 0.5

R <- matrix(0,p,p)#

for(i in 1:p){

for(j in 1:p){

R[i,j] <- rho^abs(i-j)

}

}

error=rnorm(n,0,1)#rt(n,1.5),rnorm(n,0,1),rcauchy(n,0,1)

X <- matrix(rnorm(n\*p),n,p) \%\*\% t(chol(R))#chol

X=scale(X,center=FALSE,scale=TRUE)

delte=0.5#0.5,1.5

y <- X \%\*\% beta + delte\*error

lambda=0.5\*sqrt(log(p)/n)

maxite=200

alpha=0.9

lambda1 = alpha\*lambda

lambda2 = (1-alpha)\*lambda

tau=0.5

ka=IQR(y)/10

M=1

Enetmodel=CPADMME (X,y,loss="LS",pen="SCAD",lambda1,lambda2,tau,ka,maxite,M)#loss=”LS”,”SR”,”Quan”,”Huber”; #and pen = “L1”,”SCAD”,”MCP” .

Enetmodel$K

Enetmodel$time

plot(beta)

beta[1:10]

Enetmodel$beta[1:10]

plot(Enetmodel$beta)

#######Example 2(Massive data regression)

s=0.5

n=72000

p=2560\*s

beta=rep(0,p)

for (j in 1:p) {

beta[j] = (-1)^j\*exp(-(2\*j-1)/20)

}

rho <- 0.5

R <- matrix(0,p,p)#

for(i in 1:p){

for(j in 1:p){

R[i,j] <- rho^abs(i-j)

}

}

error=rnorm(n,0,1)#rt(n,1.5),rnorm(n,0,1),rcauchy(n,0,1)

X <- matrix(rnorm(n\*p),n,p) \%\*\% t(chol(R))#chol

X=scale(X,center=FALSE,scale=TRUE)

delte=0.5#0.5,1.5

y <- X \%\*\% beta + delte\*error

lambda=0.5\*sqrt(log(p)/n)

maxite=200

alpha=0.9

lambda1 = alpha\*lambda

lambda2 = (1-alpha)\*lambda

tau=0.5

ka=IQR(y)/10

M=10#M=1,10,100

Enetmodel=CPADMME (X,y,loss="LS",pen="SCAD",lambda1,lambda2,tau,ka,maxite,M)#loss=”LS”,”SR”,”Quan”,”Huber”; #and pen = “L1”,”SCAD”,”MCP” .

Enetmodel$K

Enetmodel$time

plot(beta)

beta[1:10]

Enetmodel$beta[1:10]

plot(Enetmodel$beta)

**Two examples for sparse group regressions**

#######Example 1(High-dimensional regression)

s=1#s=1,2,3,4,5....

n=720\*10^0#\*100

p=2560\*s

index=1:80

ind=sample(index,10)

beta=rep(0,p)

for (i in 1:10) {

beta[((ind[i]-1)\*(32\*s)+1):((ind[i]-1)\*(32\*s)+(32\*s))]=rep(runif(1,-3,3),(32\*s))

}

plot(beta)

rho <- 0.5

R <- matrix(0,p,p)#

for(i in 1:p){

for(j in 1:p){

R[i,j] <- rho^abs(i-j)

}

}

error=rnorm(n,0,1)#rt(n,1.5),rnorm(n,0,1),rcauchy(n,0,1)

X <- matrix(rnorm(n\*p),n,p) \%\*\% t(chol(R))#chol

X=scale(X,center=FALSE,scale=TRUE)

delte=0.5#0.5,1.5

y <- X \%\*\% beta + delte\*error

F\_matrix<-function(p)

{

F=matrix(0,p-1,p)

for(i in 1:(p-1))

{

F[i,i]=1

F[i,i+1]=-1

}

return(F)

}

F\_m=F\_matrix(p)

#

#

FTF=function(p){

if(p==2){FTF=matrix(c(1,-1,-1,1),2,2)}else{

FTF=diag(2,p)

for (i in 2:p) {

FTF[i-1,i]=-1

FTF[i,i-1]=-1

}

FTF[1,1]=1

FTF[p,p]=1}

return(FTF)

}

FF=FTF(p)

lambda=0.5\*sqrt(log(p)/n)

alpha=0.2

lambda1 = alpha\*lambda

lambda2 = 1\*(1-alpha)\*lambda

tau=0.5

maxite = 200

ka=IQR(y)/10

M=1

fmodel = CPADMMF(X,y,loss="Quan",pen="MCP",lambda1,lambda2,tau,ka,maxite,M)

fmodel$K

fmodel$time

plot(beta)

plot(fmodel$beta)

#######Example 2(Massive data regression)

s=0.5

n=720\*10^3#\*100

p=2560\*s

index=1:80

ind=sample(index,10)

beta=rep(0,p)

for (i in 1:10) {

beta[((ind[i]-1)\*(32\*s)+1):((ind[i]-1)\*(32\*s)+(32\*s))]=rep(runif(1,-3,3),(32\*s))

}

plot(beta)

rho <- 0.5

R <- matrix(0,p,p)#

for(i in 1:p){

for(j in 1:p){

R[i,j] <- rho^abs(i-j)

}

}

error=rnorm(n,0,1)#rt(n,1.5),rnorm(n,0,1),rcauchy(n,0,1)

X <- matrix(rnorm(n\*p),n,p) \%\*\% t(chol(R))#chol

X=scale(X,center=FALSE,scale=TRUE)

delte=0.5#0.5,1.5

y <- X \%\*\% beta + delte\*error

F\_matrix<-function(p)

{

F=matrix(0,p-1,p)

for(i in 1:(p-1))

{

F[i,i]=1

F[i,i+1]=-1

}

return(F)

}

F\_m=F\_matrix(p)

#

#

FTF=function(p){

if(p==2){FTF=matrix(c(1,-1,-1,1),2,2)}else{

FTF=diag(2,p)

for (i in 2:p) {

FTF[i-1,i]=-1

FTF[i,i-1]=-1

}

FTF[1,1]=1

FTF[p,p]=1}

return(FTF)

}

FF=FTF(p)

lambda=0.5\*sqrt(log(p)/n)

alpha=0.2

lambda1 = alpha\*lambda

lambda2 = 1\*(1-alpha)\*lambda

tau=0.5

maxite = 200

ka=IQR(y)/10

M=10#M=1,10,100

fmodel = CPADMMF(X,y,loss="Quan",pen="MCP",lambda1,lambda2,tau,ka,maxite,M)

fmodel$K

fmodel$time

plot(beta)

plot(fmodel$beta)

**Two examples for sparse fused regressions**

#######Example 1(High-dimensional regression)

s=1#s=1,2,3,4,5....

n=720\*10^0

p=2560\*s

index=1:80

ind=sample(index,10)

beta=rep(0,p)

for (i in 1:10) {

for (j in 1:(32\*s)) {

betag=sign(runif(1,-1,1))\*(1+ abs(rnorm(32\*s,0,1)))

}

beta[((ind[i]-1)\*(32\*s)+1):((ind[i]-1)\*(32\*s)+(32\*s))]=betag

}

rho <- 0.5

R <- matrix(0,p,p)#

for(i in 1:p){

for(j in 1:p){

R[i,j] <- rho^abs(i-j)

}

}

error=rnorm(n,0,1)#rt(n,1.5),rnorm(n,0,1),rcauchy(n,0,1)

X <- matrix(rnorm(n\*p),n,p) \%\*\% t(chol(R))#chol

X=scale(X,center=FALSE,scale=TRUE)

delte=0.5#0.5,1.5

y <- X \%\*\% beta + delte\*error

lambda=0.5\*sqrt(log(p)/n)

maxite=200

alpha=0.9

lambda1 = alpha\*lambda

lambda2 = (1-alpha)\*lambda

tau=0.5

ka=IQR(y)/10

G=list()

for (aa in 1:80) {

G[[aa]] <- (1:p)[(p/80\*(aa-1)+1) : (p/80\*aa)]

}

M=1#M=1,10,100

gmodel=CPADMMG(X,y,G,loss="LS",pen="SCAD",lambda1,lambda2,tau,ka,maxite,M)#loss=”LS”,”SR”,”Quan”,”Huber”; #and pen = “L1”,”SCAD”,”MCP” .

gmodel$K

gmodel$time

plot(beta)

plot(gmodel$beta)

#######Example 2(Massive data regression)

s=0.5

n=720\*10^3

p=2560\*s

index=1:80

ind=sample(index,10)

beta=rep(0,p)

for (i in 1:10) {

for (j in 1:(32\*s)) {

betag=sign(runif(1,-1,1))\*(1+ abs(rnorm(32\*s,0,1)))

}

beta[((ind[i]-1)\*(32\*s)+1):((ind[i]-1)\*(32\*s)+(32\*s))]=betag

}

rho <- 0.5

R <- matrix(0,p,p)#

for(i in 1:p){

for(j in 1:p){

R[i,j] <- rho^abs(i-j)

}

}

error=rnorm(n,0,1)#rt(n,1.5),rnorm(n,0,1),rcauchy(n,0,1)

X <- matrix(rnorm(n\*p),n,p) \%\*\% t(chol(R))#chol

X=scale(X,center=FALSE,scale=TRUE)

delte=0.5#0.5,1.5

y <- X \%\*\% beta + delte\*error

lambda=0.5\*sqrt(log(p)/n)

maxite=200

alpha=0.9

lambda1 = alpha\*lambda

lambda2 = (1-alpha)\*lambda

tau=0.5

ka=IQR(y)/10

G=list()

for (aa in 1:80) {

G[[aa]] <- (1:p)[(p/80\*(aa-1)+1) : (p/80\*aa)]

}

M=10#M=1,10,100

gmodel=CPADMMG(X,y,G,loss="LS",pen="SCAD",lambda1,lambda2,tau,ka,maxite,M)#loss=”LS”,”SR”,”Quan”,”Huber”; #and pen = “L1”,”SCAD”,”MCP” .

gmodel$K

gmodel$time

plot(beta)

plot(gmodel$beta)