Package 'MSGLasso'

November 9, 2016

Type Package

Title Multivariate Sparse Group Lasso for the Multivariate Multiple
Linear Regression with an Arbitrary Group Structure

Version 2.1

Date 2016-11-7

Author
Yanming Li <`liyanmin@umich.edu`>, Bin Nan <`bnan@umich.edu`>, Ji Zhu <`jizhu@umich.edu`>

Maintainer Yanming Li <`liyanmin@umich.edu`>

Description For fitting multivariate response and multiple predictor linear regressions with an arbitrary group structure assigned on the regression coefficient matrix, using the multivariate sparse group lasso and the mixed coordinate descent algorithm.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-11-08 11:50:10

R topics documented:

`Beta.m` *A simulated Beta matrix*

Description

An example data set of a simulated 200x200 Beta matrix. It consists of 10 row block groups each containing 20 rows and 10 column blocks each containing 20 columns.

Usage

```
data(Beta.m)
```

Format

The format is: a matrix of dimension 200x200.

Author(s)

Yanming Li, Bin Nan, Ji Zhu

References

Y. Li, B. Nan and J. Zhu (2015) Multivariate sparse group lasso for the multivariate multiple linear regression with an arbitrary group structure. Biometrics. DOI: 10.1111/biom.12292

Examples

```
data(Beta.m)
```

`Cal_grpWTs` *An auxilary function calculating the group weighting matrix* `grpWTs` *required when calling the MSGlasso function.*

Description

An auxilary function calculating the group weighting matrix `grpWTs` required when calling the MSGlasso function.

Usage

```
Cal_grpWTs(P, Q, G, R, gmax, PQ.grps)
```

Arguments

`P` a positive interger indicating number of predictor variables

`Q` a positive interger indicating number of response variables

`G` a positive interger indicating number of predictor groups

`R` a positive interger indicating number of response groups

`gmax` a positive interger indicating the max number of different groups a single variable (either a predictor or response variable) belongs to.

`PQ.grps` a matrix of (p+q) by (gmax+1), with each row starting with group indicators that row variable belongs to, and followed by 999's till the row is filled.

Details

Generates the required input group weighting matrix `grpWTs` when calling the main MSGlasso function. The `grpWTs` is a g by r matrix containing the adaptive weighting scores for each group. `MSGLasso.grpWTs` use the square root of the group size (number of entries the group contains) as the weight for each group.

Value

A list with one components:

`grpWTs` the `grpWTs` matrix generated

Author(s)

Yanming Li, Bin Nan, Ji Zhu

References

Y. Li, B. Nan and J. Zhu (2015) Multivariate sparse group lasso for the multivariate multiple linear regression with an arbitrary group structure. Biometrics. DOI: 10.1111/biom.12292

Examples

```
#############################################################################
## generating the grp.WTs matrix for an overlapping group structure
#############################################################################

P <- 200
Q <- 200
G <- 10
R <- 10
gmax <- 1

GarrStarts <-c(0,20,40,60,80,100,120,140,160,180)
GarrEnds <-c(19,39,59,79,99,119,139,159,179,199)
RarrStarts <-c(0,20,40,60,80,100,120,140,160,180)
RarrEnds <-c(19,39,59,79,99,119,139,159,179,199)
```

```
tmp <- FindingPQGrps(P, Q, G, R, gmax, GarrStarts, GarrEnds, RarrStarts, RarrEnds)
PQ.grps <- tmp$PQgrps

tmp1 <- Cal_grpWTs(P, Q, G, R, gmax, PQ.grps)
grp.WTs <- tmp1$grpWTs
```

FindingGRGrps

An auxilary function calculating the containing variable index for each predictor (or response) group. It generates the required input `GR.grps` *matrix when calling the MSGlasso function.*

Description

An auxilary function calculating the containing variable index for each predictor (or response) group. It generates the required input `GR.grps` matrix when calling the MSGlasso function.

Usage

```
FindingGRGrps(P, Q, G, R, cmax, G.Starts, G.Ends, R.Starts, R.Ends)
```

Arguments

`P`	a positive interger indicating number of predictor variables
`Q`	a positive interger indicating number of response variables
`G`	a positive interger indicating number of predictor groups
`R`	a positive interger indicating number of response groups
`cmax`	a positive interger indicating the max number of variables a single group (either a predictor or response group) contains.
`G.Starts`	a vector of starting coordinates for the predictor groups.
`G.Ends`	a vector of ending coordinates for the predictor groups.
`R.Starts`	a vector of starting coordinates for the response groups.
`R.Ends`	a vector of ending coordinates for the response groups.

Details

Generates the required input `GRgrps` matrix when calling the main MSGlasso function, when user provide the starting and ending coordinates for each of the predictor and response groups. The `GRgrps` is a matrix of (g+r) by (cmax+1), with each row starting with variable indicators that row group contains, and followed by 999's till the row is filled.

Value

A list with one components:

`GRgrps`	the `GRgrps` matrix generated

Author(s)

Yanming Li, Bin Nan, Ji Zhu

References

Y. Li, B. Nan and J. Zhu (2015) Multivariate sparse group lasso for the multivariate multiple linear regression with an arbitrary group structure. Biometrics. DOI: 10.1111/biom.12292

Examples

```
###########################################################################
## generating the GR.grps matrix for an overlapping group structure
###########################################################################

P <- 200
Q <- 200
G <- 10
R <- 10
cmax <- 400

GarrStarts <-c(0,20,40,60,80,100,120,140,160,180)
GarrEnds <-c(19,39,59,79,99,119,139,159,179,199)
RarrStarts <-c(0,20,40,60,80,100,120,140,160,180)
RarrEnds <-c(19,39,59,79,99,119,139,159,179,199)

tmp <- FindingGRGrps(P, Q, G, R, cmax, GarrStarts, GarrEnds, RarrStarts, RarrEnds)
GR.grps <- tmp$GRgrps
```

FindingPQGrps *An auxilary function calculating the group attribution index for each predictor (or response) variable. It generates the required input* `PQ.grps` *matrix when calling the MSGlasso function.*

Description

An auxilary function calculating the group attribution index for each predictor (or response) variable. It generates the required input `PQ.grps` matrix when calling the MSGlasso function.

Usage

```
FindingPQGrps(P, Q, G, R, gmax, G.Starts, G.Ends, R.Starts, R.Ends)
```

Arguments

P	a positive interger indicating number of predictor variables
Q	a positive interger indicating number of response variables
G	a positive interger indicating number of predictor groups
R	a positive interger indicating number of response groups
gmax	a positive interger indicating the max number of different groups a single variable (either a predictor or response variable) belongs to.
G.Starts	a vector of starting coordinates for the predictor groups.
G.Ends	a vector of ending coordinates for the predictor groups.
R.Starts	a vector of starting coordinates for the response groups.
R.Ends	a vector of ending coordinates for the response groups.

Details

Generates the required input PQgrps matrix when calling the main MSGlasso function, when user provide the starting and ending coordinates for each of the predictor and response groups. The PQgrps is a matrix of (p+q) by (gmax+1), with each row starting with group indicators that row variable belongs to, and followed by 999's till the row is filled.

Value

A list with one components:

PQgrps	the PQgrps matrix generated

Author(s)

Yanming Li, Bin Nan, Ji Zhu

References

Y. Li, B. Nan and J. Zhu (2015) Multivariate sparse group lasso for the multivariate multiple linear regression with an arbitrary group structure. Biometrics. DOI: 10.1111/biom.12292

Examples

```
##############################################################################
## generating the PQ.grps matrix for an overlapping group structure
##############################################################################

P <- 200
Q <- 200
G <- 10
R <- 10
gmax <- 1

GarrStarts <-c(0,20,40,60,80,100,120,140,160,180)
```

```
GarrEnds <-c(19,39,59,79,99,119,139,159,179,199)
RarrStarts <-c(0,20,40,60,80,100,120,140,160,180)
RarrEnds <-c(19,39,59,79,99,119,139,159,179,199)

tmp <- FindingPQGrps(P, Q, G, R, gmax, GarrStarts, GarrEnds, RarrStarts, RarrEnds)
PQ.grps <- tmp$PQgrps
```

MSGLasso — *A function to fit the Multivariate Sparse Group Lasso with an arbitrary group structure (MSGLasso)*

Description

A function to fit the Multivariate Sparse Group Lasso with an arbitrary group structure using the mixed coordinate descent algorithm.

Usage

```
MSGLasso(X.m, Y.m, grp.WTs, Pen.L, Pen.G, PQ.grps, GR.grps, grp_Norm0,
 lam1, lam.G, Beta0 = NULL)
```

Arguments

`X.m` numeric predictor matrix (n by p): columns correspond to predictor variables and rows correspond to samples. Missing values are not allowed.

`Y.m` numeric predictor matrix (n by q): columns correspond to response variables and rows correspond to samples. Missing values are not allowed.

`grp.WTs` user specified adaptive group weighting matrix of g by r, for putting different penalization levels on different groups. Missing values are not allowed.

`Pen.L` user specified single-entry level penalization indictor mateix of p by q. 1 for being penalized and 0 for not. Missing values are not allowed.

`Pen.G` user specified group level penalization indictor mateix of g by r. 1 for being penalized and 0 for not. Missing values are not allowed.

`PQ.grps` the group attributing matrix of (p+q) by (gmax+1), where gmax is max number of different groups a single variable belongs to. Each row corresponds to a (predictor or response) varaible, and starts with group indexes the variable belongs to and followed by 999.

`GR.grps` the variable attributing matrix of (g+r)*(cmax+1), where cmax is max number of variables a single group contains. Each row corresponds to a (predictor or response) group, and starts with variable indexes the group contains to and followed by 999.

`grp_Norm0` a numeric matrix (g by r) containing starting L2 group norm values. Should be calculated from the Beta starting value matrix Beta0.

`lam1` lasso panelty parameter scaler.

`lam.G` group penalty parameter matrix (g by r).

`Beta0` numeric matrix (p by q) containing starting beta values. By defalt use a zero matrix.

Details

Uses the mixed coordinate descent algorithm for fitting the multivariate sparse group lasso in a multivariate-response-multiple-predictor linear regression setting, with an arbitrary group structure on the regression coefficient matrix (Li, Nan and Zhu 2014).

Value

A list with five components:

`Beta` the estimated regression coefficient matrix (p by q).

`grpNorm` the L2 group norm matrix (g by r) of the estimated regression coefficient matrix.

`E` residual matrix (n by q).

`rss.v` a vector of length q recording the resisual sum square for each of the q responses.

`rss` a scaler of overall residual sum of square.

`iter` a positive interger recording the number of iterations till convergence.

Author(s)

Yanming Li, Bin Nan, Ji Zhu

References

Y. Li, B. Nan and J. Zhu (2015) Multivariate sparse group lasso for the multivariate multiple linear regression with an arbitrary group structure. Biometrics. DOI: 10.1111/biom.12292

Examples

```
#####################################################
# Simulate data
#####################################################

set.seed(sample(1:100,1))
G.arr <- c(0,20,20,20,20,20,20,20,20,20,20)

data("Beta.m")

######## generate data set for model fitting

simDataGen<-function(N, Beta, rho, s, G.arr, seed=1){

P<-nrow(Beta)
Q<-ncol(Beta)
gsum<-0
X.m<-NULL
set.seed(seed)

Sig<-matrix(0,P,P)
jstart <-1

for(g in 1:length(G.arr)-1){
X.m<-cbind(X.m, matrix(rnorm(N*G.arr[g+1]),N,G.arr[g+1], byrow=TRUE))

for(i in 2:P){ for(j in jstart: (i-1)){

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

    Sig[j,i]<-Sig[i,j]

}}
jstart <- jstart + G.arr[g+1]
}


diag(Sig)<-1
R<-chol(Sig)

X.m<-X.m%*%R

SVsum <-0

for (q in 1:Q){SVsum <-SVsum+var(X.m %*% Beta[,q])}
sdr =sqrt(s*SVsum/Q)

E.m <- matrix(rnorm(N*Q,0,sdr),N, Q, byrow=TRUE)

Y.m<-X.m%*%Beta+E.m

return(list(X=X.m, Y=Y.m, E=E.m))
}

N <-150

rho=0.5;
s=4;

Data <- simDataGen(N, Beta.m,rho, s, G.arr, seed=sample(1:100,1))
X.m<-Data$X
Y.m<-Data$Y


############################################################
## fit model for one set of (lam1, lam.G) using example data
############################################################

P <- dim(Beta.m)[1]
Q <- dim(Beta.m)[2]
```

```
G <- 10
R <- 10

gmax <- 1
cmax <- 20
GarrStarts <- c(0,20,40,60,80,100,120,140,160,180)
GarrEnds <- c(19,39,59,79,99,119,139,159,179,199)
RarrStarts <- c(0,20,40,60,80,100,120,140,160,180)
RarrEnds <- c(19,39,59,79,99,119,139,159,179,199)

tmp <- FindingPQGrps(P, Q, G, R, gmax, GarrStarts, GarrEnds, RarrStarts, RarrEnds)
PQgrps <- tmp$PQgrps

tmp1 <- Cal_grpWTs(P, Q, G, R, gmax, PQgrps)
grpWTs <- tmp1$grpWTs

tmp2 <- FindingGRGrps(P, Q, G, R, cmax, GarrStarts, GarrEnds, RarrStarts, RarrEnds)
GRgrps <- tmp2$GRgrps

Pen_L <- matrix(rep(1,P*Q),P,Q, byrow=TRUE)
Pen_G <- matrix(rep(1,G*R),G,R, byrow=TRUE)
grp_Norm0 <- matrix(rep(0, G*R), nrow=G, byrow=TRUE)

MSGLassolam1 <- 1.6
MSGLassolamG <- 0.26
MSGLassolamG.m <- matrix(rep(MSGLassolamG, G*R),G,R,byrow=TRUE)

system.time(try <-MSGLasso(X.m, Y.m, grpWTs, Pen_L, Pen_G, PQgrps, GRgrps, grp_Norm0,
   MSGLassolam1, MSGLassolamG.m, Beta0=NULL))

## Not run:
#################################################
## visulizing model fitting results
#################################################

########visulizing selection effect using heatmaps

MYplotBW <- function(Beta){
colorNP <- ceiling(abs(max(Beta)))+2
ColorValueP <- colorRampPalette(c(”gray50”, ”black”))(colorNP)
colorNN <- ceiling(abs(min(Beta)))+2
ColorValueN <- colorRampPalette(c(”gray50”, ”white”))(colorNN)
P <-nrow(Beta)
Q <-ncol(Beta)
Xlim <- c(0,2*(P+1))
Ylim <- c(0,2*(Q+1))
plot(0, type=”n”, xlab=””, ylab=””, xlim=Xlim, ylim=Ylim, cex.lab=1.0,
  bty=”n”, axes=FALSE)
for (p in 1:P){
for (q in 1:Q){
        k0 <- Beta[p,q]
if(k0==0){
      rect(2*(P-p+1)-1,2*(Q-q+1)-1, 2*(P-p+1)+1, 2*(Q-q+1)+1, col=”white”, border=NA)
}
if(k0>0){
      k <- ceiling(k0)+1
      if(k>2) {k <- k+1}
          rect(2*(P-p+1)-1,2*(Q-q+1)-1, 2*(P-p+1)+1, 2*(Q-q+1)+1,
               col="black", border=NA)
}
if(k0<0){
      k <- ceiling(abs(k0))+1
      if(k>2) {k <- k+1}
          rect(2*(P-p+1)-1,2*(Q-q+1)-1, 2*(P-p+1)+1, 2*(Q-q+1)+1,
               col="black", border=NA)
}
}
}

rect(1,1,2*P, 2*Q, lty=2)
}

MYplotBW(try$Beta)

rect(1,1,40,40, lty=2)
rect(41,41,80,80, lty=2)
rect(81,81,120,120, lty=2)
rect(121,121,160,160, lty=2)
rect(161,161,200,200, lty=2)
rect(201,201,240,240, lty=2)
rect(241,241,280,280, lty=2)
rect(281,281,320,320, lty=2)
rect(361,1,400,400, lty=2)

######## visulizing the true Beta matrix

#X11()

MYplotBW(Beta.m)

rect(1,1,40,40, lty=2)
rect(41,41,80,80, lty=2)
rect(81,81,120,120, lty=2)
rect(121,121,160,160, lty=2)
rect(161,161,200,200, lty=2)
rect(201,201,240,240, lty=2)
rect(241,241,280,280, lty=2)
rect(281,281,320,320, lty=2)
rect(361,1,400,400, lty=2)


## End(Not run)
```

MSGLasso.cv *Fit the MSGLasso for a series sets of tuning parameters and use the k-fold cross validation to select the optimal tunning parameter set.*

Description

Fit the MSGLasso for a series sets of tuning parameters and use the k-fold cross validation to select the optimal tunning parameter set.

Usage

```
MSGLasso.cv(X, Y, grpWTs, Pen.L, Pen.G, PQgrps, GRgrps, lam1.v, lamG.v,
   fold = 10, seed = 1, Beta.ini = NULL, grp_Norm = NULL)
```

Arguments

`X` numeric predictor matrix (n by p): columns correspond to predictor variables and rows correspond to samples. Missing values are not allowed.

`Y` numeric predictor matrix (n by q): columns correspond to response variables and rows correspond to samples. Missing values are not allowed.

`grpWTs` user specified adaptive group weighting matrix of g by r, for putting different penalization levels on different groups. Missing values are not allowed.

`Pen.L` user specified single-entry level penalization indictor mateix of p by q. 1 for being penalized and 0 for not. Missing values are not allowed.

`Pen.G` user specified group level penalization indictor mateix of g by r. 1 for being penalized and 0 for not. Missing values are not allowed.

`PQgrps` the group attributing matrix of (p+q) by (gmax+1), where gmax is max number of different groups a single variable belongs to. Each row corresponds to a (predictor or response) varaible, and starts with group indexes the variable belongs to and followed by 999.

`GRgrps` the variable attributing matrix of (g+r)*(cmax+1), where cmax is max number of variables a single group contains. Each row corresponds to a (predictor or response) group, and starts with variable indexes the group contains to and followed by 999.

`lam1.v` lasso panelty parameter scaler.

`lamG.v` group penalty parameter matrix (g by r).

`fold` a positive integer for the corss validation fold. Default=5.

`seed` a numeric scaler, specifying the seed of the random number generator in R for generating cross validation subset for each fold. Default=1.

`Beta.ini` a numeric matrix of p by q, specifying the starting values of the input Beta matrix for each fold. Default using the zero matrix.

`grp_Norm` a numeric matrix (g by r) containing starting L2 group norm values. Should be calculated from the Beta starting value matrix Beta.ini.

Details

Performs a k-fold cross-validation for seaching the optimal tunning parameter associated with the minimal prediction error on a two-dimensional grid.

Value

A list with two components:

`rss.cv` a numeric matrix recording the cross validation scores based on the MSGLasso estimators for each pair of (lam1, lamG).

`lams.c` a list of tuning parameter pairs corresponding the validation scores in the vectorized rss.cv.

Author(s)

Yanming Li, Bin Nan, Ji Zhu

References

Y. Li, B. Nan and J. Zhu (2015) Multivariate sparse group lasso for the multivariate multiple linear regression with an arbitrary group structure. Biometrics. DOI: 10.1111/biom.12292

Examples

```
## Not run:
######################################################
# Simulate data
######################################################

set.seed(sample(1:100,1))
G.arr <- c(0,20,20,20,20,20,20,20,20,20,20)

data("Beta.m")

######## generate data set for model fitting

simDataGen<-function(N, Beta, rho, s, G.arr, seed=1){

P<-nrow(Beta)
Q<-ncol(Beta)
gsum<-0
X.m<-NULL

set.seed(seed)

Sig<-matrix(0,P,P)
jstart <-1

for(g in 1:length(G.arr)-1){
X.m<-cbind(X.m, matrix(rnorm(N*G.arr[g+1]),N,G.arr[g+1], byrow=TRUE))
for(i in 2:P){ for(j in jstart: (i-1)){

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

    Sig[j,i]<-Sig[i,j]

}}
jstart <- jstart + G.arr[g+1]
}


diag(Sig)<-1
R<-chol(Sig)

X.m<-X.m%*%R

SVsum <-0

for (q in 1:Q){SVsum <-SVsum+var(X.m %*% Beta[,q])}
sdr =sqrt(s*SVsum/Q)

E.m <- matrix(rnorm(N*Q,0,sdr),N, Q, byrow=TRUE)

Y.m<-X.m%*%Beta+E.m

return(list(X=X.m, Y=Y.m, E=E.m))
}

N <-150

rho=0.5;
s=4;

Data <- simDataGen(N, Beta.m,rho, s, G.arr, seed=sample(1:100,1))
X.m<-Data$X
Y.m<-Data$Y

################################################
## cross validation using the example data
################################################
P <- dim(Beta.m)[1]
Q <- dim(Beta.m)[2]
G <- 10
R <- 10

gmax <- 1
cmax <- 20
GarrStarts <- c(0,20,40,60,80,100,120,140,160,180)
GarrEnds <- c(19,39,59,79,99,119,139,159,179,199)
RarrStarts <- c(0,20,40,60,80,100,120,140,160,180)
RarrEnds <- c(19,39,59,79,99,119,139,159,179,199)
```

```
tmp <- FindingPQGrps(P, Q, G, R, gmax, GarrStarts, GarrEnds, RarrStarts, RarrEnds)
PQgrps <- tmp$PQgrps

tmp1 <- Cal_grpWTs(P, Q, G, R, gmax, PQgrps)
grpWTs <- tmp1$grpWTs

tmp2 <- FindingGRGrps(P, Q, G, R, cmax, GarrStarts, GarrEnds, RarrStarts, RarrEnds)
GRgrps <- tmp2$GRgrps

Pen_L <- matrix(rep(1,P*Q),P,Q, byrow=TRUE)
Pen_G <- matrix(rep(1,G*R),G,R, byrow=TRUE)
grp_Norm0 <- matrix(rep(0, G*R), nrow=G, byrow=TRUE)

lam1.v <- seq(1.0, 1.5, length=6)
lamG.v <- seq(0.19, 0.25, length=7)

try.cv<- MSGLasso.cv(X.m, Y.m, grpWTs, Pen_L, Pen_G, PQgrps, GRgrps,
   lam1.v, lamG.v, fold=5, seed=1)
MSGLassolam1 <- try.cv$lams.c[which.min(as.vector(try.cv$rss.cv))][[1]]$lam1
MSGLassolamG  <- try.cv$lams.c[which.min(as.vector(try.cv$rss.cv))][[1]]$lam3
MSGLassolamG.m <- matrix(rep(MSGLassolamG, G*R),G,R,byrow=TRUE)

system.time(try <-MSGLasso(X.m, Y.m, grpWTs, Pen_L, Pen_G, PQgrps, GRgrps,
    grp_Norm0, MSGLassolam1, MSGLassolamG.m, Beta0=NULL))


## End(Not run)
```

Index