

Package ‘msma’

September 1, 2019

Type Package

Title Multiblock Sparse Multivariable Analysis

Version 2.0

Date 2019-09-01

Author Atsushi Kawaguchi

Maintainer Atsushi Kawaguchi <kawa_a24@yahoo.co.jp>

Depends mvtnorm

Description Several functions can be used to analyze multiblock multivariable data. If the input is a single matrix, then principal components analysis (PCA) is implemented. If the input is a list of matrices, then multiblock PCA is implemented. If the input is two matrices, for exploratory and objective variables, then partial least squares (PLS) analysis is implemented. If the input is two lists of matrices, for exploratory and objective variables, then multiblock PLS analysis is implemented. Additionally, if an extra outcome variable is specified, then a supervised version of the methods above is implemented. For each method, sparse modeling is also incorporated. Functions for selecting the number of components and regularized parameters are also provided.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2019-09-01 13:10:02 UTC

R topics documented:

msma-package	2
cvmsma	2
msma	4
ncompsearch	6
predict.msma	8
regparasearch	9
simdata	11
strsimdata	11
summary.msma	12

Index**14**

msma-package

*Multiblock Sparse Matrix Analysis Package***Description**

A Package for Implementation of the method

Author(s)

Atsushi Kawaguchi. <kawa_a24@yahoo.co.jp>

References

Kawaguchi A, Yamashita F (2017). Supervised Multiblock Sparse Multivariable Analysis with Application to Multimodal Brain Imaging Genetics. *Biostatistics*, 18(4) 651-665.

See Also

[msma](#)

cvmsma

*Cross-Validation***Description**

cross-validated method to evaluate the fit of "msma".

Usage

```
cvmsma(X, Y = NULL, Z = NULL, comp = 1, lambdaX, lambdaY = NULL,
       lambdaXsup = NULL, lambdaYsup = NULL, eta = 1, type = "lasso",
       inX = NULL, inY = NULL, inXsup = NULL, inYsup = NULL, muX = 0,
       muY = 0, nfold = 5, seed = 1)
```

Arguments

X	a (list of) matrix, explanatory variable(s).
Y	a (list of) matrix, objective variable(s).
Z	a (list of) matrix, response variable(s).
comp	numeric scalar for the maximum number of componets to be considered.
lambdaX	numeric vector of regularized parameters for X with length equal to the number of blocks. If omitted, no regularization is conducted.
lambdaY	numeric vector of regularized parameters for Y with length equal to the number of blocks. If omitted, no regularization is conducted.

lambdaXsup	numeric vector of regularized parameters for the super weight of X with length equal to the number of blocks. If omitted, no regularization is conducted.
lambdaYsup	numeric vector of regularized parameters for the super weight of Y with length equal to the number of blocks. If omitted, no regularization is conducted.
eta	numeric scalar the parameter indexing the penalty family.
type	a character.
inX	a (list of) numeric vector to specify the variables of X which are always in the model.
inY	a (list of) numeric vector to specify the variables of X which are always in the model.
inXsup	a (list of) numeric vector to specify the blocks of X which are always in the model.
inYsup	a (list of) numeric vector to specify the blocks of Y which are always in the model.
muX	a numeric scalar for the weight of X for the supervised.
muY	a numeric scalar for the weight of Y for the supervised.
nfold	number of folds - default is 5.
seed	number of seed for the random number.

Details

k-fold cross-validation for msma

Value

err The mean cross-validated errors which has three elements consisting of the mean of errors for X and Y, the errors for X and for Y.

Examples

```
##### data #####
tmpdata = simdata(n = 50, rho = 0.8, Yps = c(10, 12, 15), Xps = 20, seed=1)
X = tmpdata$X; Y = tmpdata$Y

##### One Component CV #####
cv1 = cvmsma(X, Y, comp = 1, lambdaX=2, lambdaY=1:3, nfold=5, seed=1)
cv1

##### Two Component CV #####
cv2 = cvmsma(X, Y, comp = 2, lambdaX=2, lambdaY=1:3, nfold=5, seed=1)
cv2
```

msma

*Multiblock Sparse Partial Least Squares***Description**

This is a function for computing partial least squares estimators from multivariate multiblock data incorporating sparse and supervised modeling

Usage

```
msma(X, ...)

## Default S3 method:
msma(X, Y = NULL, Z = NULL, comp = 2,
     lambdaX = NULL, lambdaY = NULL, lambdaXsup = NULL,
     lambdaYsup = NULL, eta = 1, type = "lasso", inX = NULL,
     inY = NULL, inXsup = NULL, inYsup = NULL, muX = 0, muY = 0,
     defmethod = "canonical", scaling = TRUE, verbose = FALSE, ...)

## S3 method for class 'msma'
print(x, ...)

## S3 method for class 'msma'
plot(x, ...)
```

Arguments

X	a (list of) matrix, explanatory variable(s).
...	further arguments passed to or from other methods.
Y	a (list of) matrix, objective variable(s). This is optional. If no input for Y, then the PCA method is complemented.
Z	a (list of) matrix, response variable(s). This is optional. If no input for Z, then the unsupervised PLS/PCA is complemented.
comp	numeric scalar for the maximum number of componets to be considered.
lambdaX	numeric vector of regularized parameters for X with length equal to the number of blocks. If omitted, no regularization is conducted.
lambdaY	numeric vector of regularized parameters for Y with length equal to the number of blocks. If omitted, no regularization is conducted.
lambdaXsup	numeric vector of regularized parameters for the super weight of X with length equal to the number of blocks. If omitted, no regularization is conducted.
lambdaYsup	numeric vector of regularized parameters for the super weight of Y with length equal to the number of blocks. If omitted, no regularization is conducted.
eta	numeric scalar the parameter indexing the penalty family.
type	a character.

inX	a (list of) numeric vector to specify the variables of X which are always in the model.
inY	a (list of) numeric vector to specify the variables of Y which are always in the model.
inXsup	a (list of) numeric vector to specify the blocks of X which are always in the model.
inYsup	a (list of) numeric vector to specify the blocks of Y which are always in the model.
muX	a numeric scalar for the weight of X for the supervised.
muY	a numeric scalar for the weight of Y for the supervised.
defmethod	a character.
scaling	a logical.
verbose	information
x	an object of class "msma", usually, a result of a call to msma

Details

msma requires for two inputs, explanatory and objective variables.

Value

dmode	Which modes "PLS" or "PCA"
X	Scaled X which has a list form.
Y	Scaled Y which has a list form.
Xscale	Scaling information for X. The means and standard deviations for each block of X are returned.
Yscale	Scaling information for Y. The means and standard deviations for each block of Y are returned.
comp	the number of componets
wbX	block loading for X
sbX	block score for X
wbY	block loading for Y
sbY	block score for Y
ssX	super score for X
wsX	super loading for X
ssY	super score for Y
wsY	super loading for Y
nzwbX	number of nonzeros in block loading for X
nzwbY	number of nonzeros in block loading for Y
selectXnames	names of selected variables for X
selectYnames	names of selected variables for Y

Examples

```
##### data #####
tmpdata = simdata(n = 50, rho = 0.8, Yps = c(10, 12, 15), Xps = 20, seed=1)
X = tmpdata$X; Y = tmpdata$Y

##### One Component #####
fit1 = msma(X, Y, comp=1, lambdaX=2, lambdaY=1:3)
fit1

##### Two Component #####
fit2 = msma(X, Y, comp=2, lambdaX=2, lambdaY=1:3)
fit2

##### Matrix data #####
sigma = matrix(0.8, 10, 10)
diag(sigma) = 1
X2 = rmvnorm(50, rep(0, 10), sigma)
Y2 = rmvnorm(50, rep(0, 10), sigma)

fit3 = msma(X2, Y2, comp=1, lambdaX=2, lambdaY=2)
fit3

##### Sparse Principal Component Analysis #####
fit5 = msma(X2, comp=5, lambdaX=2.5)
summary(fit5)
```

ncompsearch

Search for Number of Components

Description

cross-validated method to determine the number of components.

Usage

```
ncompsearch(X, Y = NULL, Z = NULL, comps = 1:3, lambdaX = NULL,
  lambdaY = NULL, lambdaXsup = NULL, lambdaYsup = NULL, eta = 1,
  type = "lasso", inX = NULL, inY = NULL, inXsup = NULL,
  inYsup = NULL, muX = 0, muY = 0, nfold = 5, regpara = FALSE,
  maxrep = 3, method = c("CV", "BIC")[1], whichselect = NULL)

## S3 method for class 'ncompsearch'
print(x, ...)

## S3 method for class 'ncompsearch'
plot(x, ...)
```

Arguments

X	a (list of) matrix, explanatory variable(s).
Y	a (list of) matrix, objective variable(s).
Z	a (list of) matrix, response variable(s).
comps	numeric vector for the maximum numbers of componets to be considered.
lambdaX	numeric vector of regularized parameters for X with length equal to the number of blocks. If omitted, no regularization is conducted.
lambdaY	numeric vector of regularized parameters for Y with length equal to the number of blocks. If omitted, no regularization is conducted.
lambdaXsup	numeric vector of regularized parameters for the super weight of X with length equal to the number of blocks. If omitted, no regularization is conducted.
lambdaYsup	numeric vector of regularized parameters for the super weight of Y with length equal to the number of blocks. If omitted, no regularization is conducted.
eta	numeric scalar the parameter indexing the penalty family.
type	a character.
inX	a (list of) numeric vector to specify the variables of X which are always in the model.
inY	a (list of) numeric vector to specify the variables of X which are always in the model.
inXsup	a (list of) numeric vector to specify the blocks of X which are always in the model.
inYsup	a (list of) numeric vector to specify the blocks of Y which are always in the model.
muX	a numeric scalar for the weight of X for the supervised.
muY	a numeric scalar for the weight of Y for the supervised.
ifold	number of folds - default is 5.
regpara	logical
maxrep	numeric scalar for the number of iteration.
method	a character, the evaluation criterion, "CV" for cross-validation, based on a matrix element-wise error, and "BIC" for Bayesian information criteria. The "BIC" is the default.
whichselect	which blocks selected.
x	an object of class "ncompsearch", usually, a result of a call to ncompsearch
...	further arguments passed to or from other methods.

Details

k-fold cross-validation for msma

Value

comps	numbers of components
mincriterion	minimum criterion values
criteria	criterion values
optncomp	optimal number of components based on minimum cross-validation error

Examples

```
##### data #####
tmpdata = simdata(n = 50, rho = 0.8, Yps = c(10, 12, 15), Xps = 20, seed=1)
X = tmpdata$X; Y = tmpdata$Y

##### number of components search #####
ncomp1 = ncompsearch(X, Y, comps = c(1, 5, 10*(1:2)), nfold=5)
plot(ncomp1)
```

predict.msma

Prediction

Description

predict method for class "msma".

Usage

```
## S3 method for class 'msma'
predict(object, newX, newY = NULL, ...)
```

Arguments

object	an object of class "msma", usually, a result of a call to msma
newX	a matrix in which to look for variables with which to predict X.
newY	a matrix in which to look for variables with which to predict Y.
...	further arguments passed to or from other methods.

Details

This function provide the p value for the hypothesis test of coefficient in the model of weighted least squares method.

Value

X	predicted X
Y	predicted Y

Examples

```
##### data #####
tmpdata = simdata(n = 50, rho = 0.8, Yps = c(10, 12, 15), Xps = 20, seed=1)
X = tmpdata$X; Y = tmpdata$Y

##### Two Component #####
fit2 = msma(X, Y, comp=2, lambdaX=2, lambdaY=1:3)
summary(fit2)

##### Predict #####
test = predict(fit2, newX=X, newY=Y)
```

regparasearch	<i>Regularized Parameters Search</i>
---------------	--------------------------------------

Description

Regularized parameters search method for "msma".

Usage

```
regparasearch(X, Y = NULL, Z = NULL, eta = 1, type = "lasso",
  inX = NULL, inY = NULL, inXsup = NULL, inYsup = NULL, muX = 0,
  muY = 0, comp = 1, nfold = 5, maxrep = 3, minpct = 0,
  maxpct = 1, method = c("CV", "BIC")[1], whichselect = NULL,
  homo = NULL)

## S3 method for class 'regparasearch'
print(x, ...)
```

Arguments

X	a (list of) matrix, explanatory variable(s).
Y	a (list of) matrix, objective variable(s).
Z	a (list of) matrix, response variable(s).
eta	numeric scalar the parameter indexing the penalty family.
type	a character.
inX	a (list of) numeric vector to specify the variables of X which are always in the model.
inY	a (list of) numeric vector to specify the variables of X which are always in the model.
inXsup	a (list of) numeric vector to specify the blocks of X which are always in the model.

inYsup	a (list of) numeric vector to specify the blocks of Y which are always in the model.
muX	a numeric scalar for the weight of X for the supervised.
muY	a numeric scalar for the weight of Y for the supervised.
comp	numeric scalar for the maximum number of componets to be considered.
nfold	number of folds - default is 5.
maxrep	numeric scalar for the number of iteration.
minpct	percent of minimum candidate parameters.
maxpct	percent of maximum candidate parameters.
method	a character, the evaluation criterion, "CV" for cross-validation, based on a matrix element-wise error, and "BIC" for Bayesian information criteria. The "BIC" is the default.
whichselect	which blocks selected.
homo	same parameters.
x	an object of class "regparasearch", usually, a result of a call to regparasearch
...	further arguments passed to or from other methods.

Details

Regularized parameters search for msma

Value

optlambdaX	Optimal parameters for X
optlambdaY	Optimal parameters for Y
mincriterion	Minimum of criterion values
criteria	Resulting criterion value
pararange	Range of candidates parameters

Examples

```
##### data #####
tmpdata = simdata(n = 50, rho = 0.8, Yps = c(10, 12, 15), Xps = c(20, 15), seed=1)
X = tmpdata$X; Y = tmpdata$Y

##### Regularized parameters search #####
opt1 = regparasearch(X, Y, comp=1, method="BIC", maxrep=2, whichselect=c("X", "Y", "Xsup", "Ysup"))
opt1
fit4 = msma(X, Y, comp=1, lambdaX=opt1$optlambdaX, lambdaY=opt1$optlambdaY,
lambdaXsup=opt1$optlambdaXsup, lambdaYsup=opt1$optlambdaYsup)
fit4
summary(fit4)
```

simdata	<i>Simulate Data sets</i>
---------	---------------------------

Description

This is a function for generating multiblock data based on the multivariable normal distribution

Usage

```
simdata(n = 100, rho = 0.8, Yps = c(100, 120, 150), Xps = 500,
        seed = 1)
```

Arguments

n	a numeric scalar, sample size.
rho	a numeric scalar, correlation coefficient.
Yps	a numeric vector, numbers of columns for Y. The length of vector corresponds to the number of blocks.
Xps	a numeric vector, numbers of columns for X. The length of vector corresponds to the number of blocks.
seed	a seed number for generating random numbers.

Details

The output is a list of matrices.

Value

X	Simulated X which has a list form
Y	Simulated Y which has a list form

strsimdata	<i>Structured Simulate Data sets</i>
------------	--------------------------------------

Description

This is a function for generating multiblock data based on the multivariable normal distribution

Usage

```
strsimdata(n = 100, ncomp = 5, Xps = 10, Yps = FALSE, rho = 0.8,
           Z = FALSE, seed = 1, minpct = 0.25, maxpct = 0.75)
```

Arguments

n	a numeric scalar, sample size.
ncomp	number of components
Xps	a numeric vector, numbers of columns for X. The length of vector corresponds to the number of blocks.
Yps	a numeric vector, numbers of columns for Y. The length of vector corresponds to the number of blocks.
rho	a numeric, correlation
Z	outcome
seed	a seed number for generating random numbers.
minpct	minimum percent of nonzero
maxpct	maximum percent of nonzero

Details

The output is a list of matrices.

Value

X	Simulated X which has a list form
Y	Simulated Y which has a list form

summary.msma

Summarizing Fits

Description

summary method for class "msma".

Usage

```
## S3 method for class 'msma'
summary(object, ...)

## S3 method for class 'summary.msma'
print(x, ...)
```

Arguments

object, x	an object of class "msma", usually, a result of a call to msma
...	further arguments passed to or from other methods.

Details

This function provide the summary of results .

Examples

```
##### data #####  
tmpdata = simdata(n = 50, rho = 0.8, Yps = c(10, 12, 15), Xps = 20, seed=1)  
X = tmpdata$X; Y = tmpdata$Y  
  
##### One Component #####  
fit1 = msma(X, Y, comp=1, lambdaX=2, lambdaY=1:3)  
summary(fit1)
```

Index

*Topic **documentation**

msma-package, [2](#)

cvmsma, [2](#)

msma, [2](#), [4](#), [5](#), [8](#), [12](#)

msma-package, [2](#)

ncompsearch, [6](#), [7](#)

plot.msma (msma), [4](#)

plot.ncompsearch (ncompsearch), [6](#)

predict.msma, [8](#)

print.msma (msma), [4](#)

print.ncompsearch (ncompsearch), [6](#)

print.regparasearch (regparasearch), [9](#)

print.summary.msma (summary.msma), [12](#)

regparasearch, [9](#), [10](#)

simdata, [11](#)

strsimdata, [11](#)

summary.msma, [12](#)