Package 'msma'

May 4, 2018

Type Package

Title Multiblock Sparse Multivariable Analysis

Version 1.1

Date 2018-05-01

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)

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-04 03:50:13 UTC

R topics documented:

`msma-package` *Multiblock Sparse Multivariable Analysis Package*

Description

A Package for implementation of multiblock multivariable data analysis.

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,
  eta = 1, type = "lasso", inX = NULL, inY = NULL, muX = 0, muY = 0,
  nfold = 5, seed = 1, intseed = 1)
```

Arguments

X	a matrix or list of matrices indicating the explanatory variable(s). This parameter is required.
Y	a matrix or list of matrices indicating objective variable(s). This is optional. If there is no input for Y, then PCA is implemented.
Z	a vector, response variable(s) for implementing the supervised version of (multi-block) PCA or PLS. This is optional. The length of Z is the number of subjects. If there is no input for Z, then unsupervised PLS/PCA is implemented.
comp	numeric scalar for the number of components to be considered.
lambdaX	numeric vector of regularized parameters for X, with a length equal to the number of blocks. If lambdaX is omitted, no regularization is conducted.
lambdaY	numeric vector of regularized parameters for Y, with a length equal to the number of blocks. If lambdaY is omitted, no regularization is conducted.
eta	numeric scalar indicating the parameter indexing the penalty family. This version contains only choice 1.
type	a character, indicating the penalty family. In this version, only one choice is available: "lasso."
inX	a vector or list of numeric vectors specifying the variables in X, always included in the model
inY	a vector or list of numeric vectors specifying the variables in Y, always included in the model
muX	a numeric scalar for the weight of X for the supervised case. 0 <= muX <= 1.
muY	a numeric scalar for the weight of Y for the supervised case. 0 <= muY <= 1.
nfold	number of folds - default is 5.
seed	seed number for the random number in the cross-validation.
intseed	seed number for the random number in the parameter estimation algorithm.

Details

k-fold cross-validation for msma. The evaluation is based on the matrix element-wise errors.

Value

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

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 Multivariable Analysis*

Description

This is a function for a matrix decomposition method incorporating sparse and supervised modeling for a multiblock multivariable data analysis

Usage

```
msma(X, ...)

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

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

Arguments

X	a matrix or list of matrices indicating the explanatory variable(s). This parameter is required.
...	further arguments passed to or from other methods.
Y	a matrix or list of matrices indicating objective variable(s). This is optional. If there is no input for Y, then PCA is implemented.
Z	a vector, response variable(s) for implementing the supervised version of (multiblock) PCA or PLS. This is optional. The length of Z is the number of subjects. If there is no input for Z, then unsupervised PLS/PCA is implemented.
comp	numeric scalar for the number of components to be considered.
lambdaX	numeric vector of regularized parameters for X, with a length equal to the number of blocks. If lambdaX is omitted, no regularization is conducted.
lambdaY	numeric vector of regularized parameters for Y, with a length equal to the number of blocks. If lambdaY is omitted, no regularization is conducted.
eta	numeric scalar indicating the parameter indexing the penalty family. This version contains only choice 1.

type	a character, indicating the penalty family. In this version, only one choice is available: "lasso."
inX	a vector or list of numeric vectors specifying the variables in X, always included in the model
inY	a vector or list of numeric vectors specifying the variables in Y, always included in the model
muX	a numeric scalar for the weight of X for the supervised case. 0 <= muX <= 1.
muY	a numeric scalar for the weight of Y for the supervised case. 0 <= muY <= 1.
defmethod	a character representing the deflation method. This version has only the choice "canonical."
scaling	a logical, indicating whether or not data scaling is performed. The default is TRUE.
verbose	information
intseed	seed number for the random number in the parameter estimation algorithm.
x	an object of class "msma." Usually, a result of a call to msma

Details

msma requires at least one input X (a matrix or list). In this case, (multiblock) PCA is conducted. If Y is also specified, then a PLS is conducted using X as explanatory variables and Y as objective variables. This function scales each data matrix to a mean of 0 and variance of 1 in the default. The block structure can be represented as a list. If Z is also specified, a supervised version is implemented, and the degree is controlled by muX or muY, where 0 <= muX <= 1, 0 <= muY <= 1, and 0 <= muX + muY < 1. If a positive lambdaX or lambdaY is specified, then a sparse estimation based on the L1 penalty is implemented.

Value

dmode	Indicates mode "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 mean and standard deviation values for each block of X are returned.
Yscale	Scaling information for Y. The mean and standard deviation values for each block of Y are returned.
comp	Number of components
wbX	Block loading for X. The list has the same length as that of the input list X (number of blocks) and consists of a matrix. The number of variables is present in the row and the number of components is present in the column.
sbX	Block score for X. The list has the same length as that of the input list X (number of blocks) and consists of a matrix, with the number of subjects in the row and the number of components in the column.
wbY	Block loading for Y. The list has same length as that of the input list Y (number of blocks) and consists of a matrix, with the number of variables in the row and the number of components in the column.

sbY	Block score for Y. The list has same length as that of the input list Y (number of blocks) and consists of a matrix, with the number of subjects in the row and the number of components in the column.
ssX	Super score for X. In the matrix, the number of subjects is in the row and the number of components is in the column.
wsX	Super loading for X. In the matrix, the number of blocks is in the row and the number of components is in the column.
ssY	Super score for Y. In the matrix, the number of subjects is in the row and the number of components is in the column.
wsY	Super loading for Y. In the matrix, the number of blocks is in the row and the number of components is in the column.
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. This returns from the original names of X
selectYnames	Names of selected variables for Y. This returns from the original names of 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

Determination of the number of components based on cross-validated method or Bayesian information criterion (BIC)

Usage

```
ncompsearch(X, Y = NULL, Z = NULL, comps = 1:3, lambdaX = NULL,
  lambdaY = NULL, eta = 1, type = "lasso", inX = NULL, inY = NULL,
  muX = 0, muY = 0, nfold = 5, regpara = FALSE, maxrep = 3,
  minpct = 0, maxpct = 1, criterion = c("BIC", "CV")[1], intseed = 1)

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

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

Arguments

X	a matrix or list of matrices indicating the explanatory variable(s). This parameter is required.
Y	a matrix or list of matrices indicating objective variable(s). This is optional. If there is no input for Y, then PCA is implemented.
Z	a vector, response variable(s) for implementing the supervised version of (multiblock) PCA or PLS. This is optional. The length of Z is the number of subjects. If there is no input for Z, then unsupervised PLS/PCA is implemented.
comps	numeric vector for the candidates for the numbers of components to be selected.
lambdaX	numeric vector of regularized parameters for X, with a length equal to the number of blocks. If lambdaX is omitted, no regularization is conducted.
lambdaY	numeric vector of regularized parameters for Y, with a length equal to the number of blocks. If lambdaY is omitted, no regularization is conducted.
eta	numeric scalar indicating the parameter indexing the penalty family. This version contains only choice 1.
type	a character, indicating the penalty family. In this version, only one choice is available: "lasso."
inX	a vector or list of numeric vectors specifying the variables in X, always included in the model
inY	a vector or list of numeric vectors specifying the variables in Y, always included in the model
muX	a numeric scalar for the weight of X for the supervised case. 0 <= muX <= 1.

muY	a numeric scalar for the weight of Y for the supervised case. 0 <= muY <= 1.
nfold	number of folds - default is 5.
regpara	logical, If TRUE, the regularized parameters search is also conducted simultaneously.
maxrep	numeric scalar for the number of iteration.
minpct	minimum candidate parameters defined as a percentile of automatically determined (possible) candidates.
maxpct	maximum candidate parameters defined as a percentile of automatically determined (possible) candidates.
criterion	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.
intseed	seed number for the random number in the parameter estimation algorithm.
x	an object of class "ncompsearch", usually, a result of a call to ncompsearch
...	further arguments passed to or from other methods.

Details

This function searches for the optimal number of components.

Value

comps	numbers of components
mincriterion	minimum criterion value
criteria	criterion values
optncomp	optimal number of components with the minimum criteria value

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*(1:5)), nfold=5)
plot(ncomp1)
```

optparasearch *Parameters Search*

Description

Combined method for optimizing the number of components and regularized parameters for "msma".

Usage

```
optparasearch(X, Y = NULL, Z = NULL, search.method = c("simultaneous",
  "regpara1st", "ncomp1st", "regparaonly")[1], eta = 1, type = "lasso",
  inX = NULL, inY = NULL, muX = 0, muY = 0, comp = 1, nfold = 5,
  maxrep = 3, minpct = 0, maxpct = 1, criterion = c("BIC", "CV")[1],
  intseed = 1)

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

Arguments

X	a matrix or list of matrices indicating the explanatory variable(s). This parameter is required.
Y	a matrix or list of matrices indicating objective variable(s). This is optional. If there is no input for Y, then PCA is implemented.
Z	a vector, response variable(s) for implementing the supervised version of (multi-block) PCA or PLS. This is optional. The length of Z is the number of subjects. If there is no input for Z, then unsupervised PLS/PCA is implemented.
search.method	a character indicationg search methods, see Details. Default is "simultaneous".
eta	numeric scalar indicating the parameter indexing the penalty family. This version contains only choice 1.
type	a character, indicating the penalty family. In this version, only one choice is available: "lasso."
inX	a vector or list of numeric vectors specifying the variables in X, always included in the model
inY	a vector or list of numeric vectors specifying the variables in Y, always included in the model
muX	a numeric scalar for the weight of X for the supervised case. 0 <= muX <= 1.
muY	a numeric scalar for the weight of Y for the supervised case. 0 <= muY <= 1.
comp	numeric scalar for the number of components to be considered or the maximum canditate number of components.
nfold	number of folds - default is 5.
maxrep	numeric scalar for the number of iteration.

minpct	minimum candidate parameters defined as a percentile of automatically determined (possible) candidates.
maxpct	maximum candidate parameters defined as a percentile of automatically determined (possible) candidates.
criterion	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.
intseed	seed number for the random number in the parameter estimation algorithm.
x	an object of class "optparasearch", usually, a result of a call to optparasearch
...	further arguments passed to or from other methods.

Details

A function for identifying the regularized sparseness parameters lambdaX and lambdaY and the number of components for msma. Four search methods are available. The "simultaneous" method identifies the number of components by searching the regularized parameters in each component. The "regpara1st" identifies the regularized parameters by fixing the number of components, then searching for the number of components with the selected regularized parameters. The "ncomp1st" method identifies the number of components with a regularized parameter of 0, then searches for the regularized parameters with the selected number of components. The "regparaonly" method searches for the regularized parameters with a fixed number of components.

Value

optncomp	Optimal number of components
optlambdaX	Optimal parameters for X
optlambdaY	Optimal parameters for Y
mincriterion	Minimum criterion value
criteria	All resulting criterion values in the process
pararange	Range of candidates parameters

Examples

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

##### Regularized parameters search #####
opt1 = optparasearch(X, Y, search.method = "regparaonly", comp=1, nfold=5, maxrep=2)
opt1
fit4 = msma(X, Y, comp=opt1$optncomp, lambdaX=opt1$optlambdaX, lambdaY=opt1$optlambdaY)
fit4
summary(fit4)

##### Restrict search range #####
opt2 = optparasearch(X, Y, comp=3, nfold=5, maxrep=2, minpct=0.5)
opt2
```

plot.msma *Plot msma*

Description

plot method for class "msma".

Usage

```
## S3 method for class 'msma'
plot(x, v = c("weight", "score", "cpev")[1], axes = 1,
  block = c("block", "super")[1], plottype = c("bar", "scatter")[1],
  XY = c("X", "Y", "XY")[1], col = NULL, signflip = FALSE, ...)
```

Arguments

x	an object of class "msma." Usually, a result of a call to msma
v	a character, "weight" for the weight, "score" for the score, and "cpev" for the cumulative percentage of explained variance (CPEV) .
axes	a numeric (or vector), specifying the component(s) to plot.
block	a character, indicating which the "block" or "super" is used.
plottype	a character, indicating the plot type. "bar" for the bar plot, "scatter" for the scatter plot.
XY	a character, indicating "X" or "Y". "XY" for the scatter plots using X and Y scores from PLS.
col	a color vector.
signflip	a logical if the sign in the block is flipped to pose the super as possitive.
...	further arguments passed to or from other methods.

Details

This function provides a plot of results.

Examples

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

fit1 = msma(X, Y, comp=1, lambdaX=2, lambdaY=1:3)
plot(fit1)
```

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 the variables used to predict X. This is required.
newY	a matrix in which to look for the variables used to predict Y.
...	further arguments passed to or from other methods.

Details

This function produces a prediction from new data based on msma fit. It is mainly used in cross-validation

Value

X	predicted X
sbX	block score for X
Y	predicted Y
sbY	block score 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

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

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

regparasearch *Regularized Parameters Search*

Description

Regularized parameters search method for "msma".

Usage

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

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

Arguments

X	a matrix or list of matrices indicating the explanatory variable(s). This parameter is required.
Y	a matrix or list of matrices indicating objective variable(s). This is optional. If there is no input for Y, then PCA is implemented.
Z	a vector, response variable(s) for implementing the supervised version of (multi-block) PCA or PLS. This is optional. The length of Z is the number of subjects. If there is no input for Z, then unsupervised PLS/PCA is implemented.
eta	numeric scalar indicating the parameter indexing the penalty family. This version contains only choice 1.
type	a character, indicating the penalty family. In this version, only one choice is available: "lasso."
inX	a vector or list of numeric vectors specifying the variables in X, always included in the model
inY	a vector or list of numeric vectors specifying the variables in Y, always included in the model
muX	a numeric scalar for the weight of X for the supervised case. 0 <= muX <= 1.
muY	a numeric scalar for the weight of Y for the supervised case. 0 <= muY <= 1.
comp	numeric scalar for the number of components to be considered.
nfold	number of folds. Default is 5.
maxrep	numeric scalar for the number of iterations.
minpct	minimum candidate parameters defined as a percentile of automatically determined (possible) candidates.
maxpct	maximum candidate parameters defined as a percentile of automatically determined (possible) candidates.

criterion	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.
intseed	seed number for the random number in the parameter estimation algorithm.
x	an object of class "regparasearch", usually, a result of a call to regparasearch
...	further arguments passed to or from other methods.

Details

This is a function for identifying the regularized parameters of sparseness lambdaX and lambdaY for msma. The initial range of candidates is computed based on fit, with regularized parameter values of 0. A binary search is conducted for dividing the parameter range into two regions. The representative value for the region is a median value, and the optimal region is selected using the minimum criteria obtained from the fit with that median value. The CV error or BIC can be used as criteria. The selected region is also divided into two region and the same process is iterated by maxrep times. Thus, the final median value in the selected region is set to be the optimal regularized parameter. The search is conducted with combinations of parameters for X and Y. The range of candidates for regularized parameters can be restricted, with a percentile of the limit (minimum or maximum) for the range.

Value

optlambdaX	Optimal parameters for X
optlambdaY	Optimal parameters for Y
mincriterion	Minimum criterion value
criteria	All resulting criterion values in the process
pararange	Range of candidates parameters

Examples

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

##### Regularized parameters search #####
opt1 = regparasearch(X, Y, comp=1, nfold=5, maxrep=2)
opt1
fit4 = msma(X, Y, comp=1, lambdaX=opt1$optlambdaX, lambdaY=opt1$optlambdaY)
fit4
summary(fit4)

##### Restrict search range #####
opt2 = regparasearch(X, Y, comp=1, nfold=5, maxrep=2, minpct=0.5)
opt2
```

simdata *Generate Test Data Sets*

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 for sample size.
rho	a numeric scalar. Correlation coefficient for all matrices.
Yps	a numeric vector indicating the numbers of columns for Y. The length of the vector corresponds to the number of blocks.
Xps	a numeric vector indicating the numbers of columns for X. The length of the vector corresponds to the number of blocks.
seed	a seed number for generating random numbers for reproducibility. Should be changed in an iterative study.

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 provides a 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

fit1 = msma(X, Y, comp=1, lambdaX=2, lambdaY=1:3)
summary(fit1)
```

Index