Package 'prinvars'

January 9, 2023

Type Package

Title Principal Variables

Version 1.0.0

Description Provides methods for reducing the number of features within a data set. See Bauer JO (2021) <doi:10.1145/3475827.3475832> and Bauer JO, Drabant B (2021) <doi:10.1016/j.jmva.2021.104754> for more information on principal loading analysis.

License MIT + file LICENSE

Imports methods, Rdpack, elasticnet, PMA

RdMacros Rdpack

Encoding UTF-8

URL https://github.com/Ronho/prinvars

BugReports https://github.com/Ronho/prinvars/issues

RoxygenNote 7.2.1

Collate 'block.R' 'cor.R' 'explained-variance.R' 'get-blocks.R' 'thresholding.R' 'scale.R' 'utils.R' 'pla.R' 'prinvars-package.R'

Suggests testthat (>= 3.0.0), AER

Config/testthat/edition 3

NeedsCompilation no

Author Jan O. Bauer [aut], Ron Holzapfel [aut, cre]

Maintainer Ron Holzapfel <ronholzapfel@outlook.de>

Repository CRAN

Date/Publication 2023-01-09 09:50:09 UTC

R topics documented:

`Block-class` *Block*

Description

Class used within the package to keep the structure and information about the generated blocks.

Slots

`features` a vector of numeric which contains the indices of the block.

`explained_variance` a numeric which contains the variance explained of the blocks variables based on the whole data set.

`is_valid` a logical which indicates if the block structure is valid.

`ev_influenced` a vector of numeric which contains the indices of the eigenvectors influenced by this block.

`pla` *Principal Loading Analysis*

Description

This function performs a principal loading analysis on the given data matrix.

Usage

```
pla(
  x,
  cor = FALSE,
  scaled_ev = FALSE,
  thresholds = 0.33,
  threshold_mode = c("cutoff", "percentage"),
  expvar = c("approx", "exact"),
  check = c("rnc", "rows"),
  ...
)
```

Arguments

x	a numeric matrix or data frame which provides the data for the principal loading analysis.
cor	a logical value indicating whether the calculation should use the correlation or the covariance matrix.
scaled_ev	a logical value indicating whether the eigenvectors should be scaled.
thresholds	a numeric value or list of numeric values used to determine "small" values inside the eigenvectors. If multiple values are given, a list of pla results will be returned.
threshold_mode	a character string indicating how the threshold is determined and used. `cutoff` indicates the usage of a threshold value. `percentage` indicates that the cutoff value is determined by the maximum element of each vector multiplied with the threshold value.
expvar	a character string indicating the method used for calculating the explained variance. `approx` uses the explained variance of each eigenvector i.e. its eigenvalue. `exact` uses the variance of each variable.
check	a character string indicating if only rows or rows as well as columns are used to detect the underlying block structure. `rows` checks if the rows fulfill the required structure. `rnc` checks if rows and columns fulfill the required structure.
...	further arguments passed to or from other methods.

Value

single or list of `pla` class containing the following attributes:

x	a numeric matrix or data frame which equals the input of `x`.
c	a numeric matrix or data frame which is the covariance or correlation matrix based on the input of `cov`.
loadings	a matrix of variable loadings (i.e. a matrix containing the eigenvectors of the dispersion matrix).
threshold	a numeric value which equals the input of `thresholds`.
threshold_mode	a character string which equals the input of `threshold_mode`.
blocks	a list of blocks which are identified by principal loading analysis.

See Bauer and Drabant (2021) for more information.

References

Bauer JO, Drabant B (2021). "Principal loading analysis." *Journal of Multivariate Analysis*, **184**, 104754. ISSN 0047259X, doi:10.1016/j.jmva.2021.104754.

Examples

```
if(requireNamespace("AER")){
require(AER)
data("OECDGrowth")

## The scales in OECDGrowth differ hence using the correlation matrix is
## highly recommended.

pla(OECDGrowth, thresholds=0.5) ## not recommended
pla(OECDGrowth, cor=TRUE, thresholds=0.5)

## We obtain three blocks: (randd), (gdp85, gdp60) and (invest, school,
## popgrowth). Block 1, i.e. the 1x1 block (randd), explains only 5.76% of
## the overall variance. Hence, discarding this block seems appropriate.

pla_obj = pla(OECDGrowth, cor=TRUE, thresholds=0.5)
pla.drop_blocks(pla_obj, c(1)) ## drop block 1

## Sometimes, considering the blocks we keep rather than the blocks we want
## to discard might be more convenient.

pla.keep_blocks(pla_obj, c(2,3)) ## keep block 2 and block 3
}
```

`pla.drop_blocks` *Drop Blocks*

Description

Used to pass the indices of the blocks we want to discard.

Usage

```
pla.drop_blocks(object, blocks, ...)
```

Arguments

`object`	a pla object.
`blocks`	a list of numeric values indicating the indices of the blocks that should be removed.
`...`	further arguments passed to or from other methods.

Value

list of the following attributes:

`x`	a numeric matrix or data frame containing the reduced set of original variables.

cc_matrix	a numeric matrix or data frame which contains the conditional dispersion matrix. Depending on the pla procedure, this is either the conditional covariance matrix or the conditional correlation matrix.

Examples

```
if(requireNamespace("AER")){
require(AER)
data("OECDGrowth")

pla(OECDGrowth, cor=TRUE, thresholds=0.5)

## we obtain three blocks: (randd), (gdp85,gdp60) and (invest, school,
## popgrowth). Block 1, i.e. the 1x1 block (randd), explains only 5.76% of
## the overall variance. Hence, discarding this block seems appropriate.

pla_obj = pla(OECDGrowth, cor=TRUE, thresholds=0.5)
pla.drop_blocks(pla_obj, c(1)) ## drop block 1
}
```

---

pla.keep_blocks *Keep Blocks*

Description

Used to pass the indices of the blocks we want to keep (i.e. which we do no want to be discarded).

Usage

```
pla.keep_blocks(object, blocks, ...)
```

Arguments

object	a pla object.
blocks	a list of numeric values indicating the indices of the blocks that should be kept.
...	further arguments passed to or from other methods.

Value

list of the following attributes:

x	a numeric matrix or data frame containing the reduced set of original variables.
cc_matrix	a numeric matrix or data frame which contains the conditional dispersion matrix. Depending on the pla procedure, this is either the conditional covariance matrix or the conditional correlation matrix.

Examples

```
if(requireNamespace("AER")){
require(AER)
data("OECDGrowth")

pla(OECDGrowth, cor=TRUE, thresholds=0.5)

## we obtain three blocks: (randd), (gdp85,gdp60) and (invest, school,
## popgrowth). Block 1, i.e. the 1x1 block (randd), explains only 5.76% of
## the overall variance. Hence, discarding this block seems appropriate.
## Therefore, we keep block 2 and block 3.

pla_obj = pla(OECDGrowth, cor=TRUE, thresholds=0.5)
pla.keep_blocks(pla_obj, c(2,3)) ## keep block 2 and block 3
}
```

`print.pla` *Print Function for pla S3*

Description

Prints the blocks, threshold, threshold_mode and the loadings.

Usage

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

Arguments

`x`	a pla object.
`...`	further arguments passed to or from other methods.

Value

A pla object which equals the input of `x`.

Examples

```
if(requireNamespace("AER")){
require(AER)
data("OECDGrowth")

pla_obj = pla(OECDGrowth, cor=TRUE, thresholds=0.5)
print(pla_obj)
}
```

show,Block-method *Block - Show*

Description

Prints the blocks structure.

Usage

```
## S4 method for signature 'Block'
show(object)
```

Arguments

object	block.

Value

No return value.

Examples

```
block <- new("Block", features=c(2, 5), explained_variance=0.03)
print(block)
```

spla *Sparse Principal Loading Analysis*

Description

This function performs sparse principal loading analysis on the given data matrix. We refer to Bauer (2022) for more information. The corresponding sparse loadings are calculated either using PMD from the PMA package or using spca from the elasticnet package. The respective methods are given by Zou et al. (2006) and Witten et al. (2009) respectively.

Usage

```
spla(
  x,
  method = c("pmd", "spca"),
  para,
  cor = FALSE,
  criterion = c("corrected", "normal"),
  threshold = 1e-07,
  rho = 1e-06,
  max.iter = 200,
```

```
  trace = FALSE,
  eps.conv = 0.001,
  orthogonal = TRUE,
  check = c("rnc", "rows"),
  ...
)
```

Arguments

x	a numeric matrix or data frame which provides the data for the sparse principal loading analysis.
method	chooses the methods to calculate the sparse loadings. pmd uses the method from Witten et al. (2009) and spca uses the method from Zou et al. (2006).
para	when method="pmd": an integer giving the bound for the L1 regularization. When method="spca": a vector containing the regularization parameter for each variable.
cor	a logical value indicating whether the calculation should use the correlation or the covariance matrix.
criterion	a character string indicating if the weight-corrected evaluation criterion (CEC) or the evaluation criterion (EC) is used. corrected changes the loadings to weight all variables equally while normal does not change the loadings.
threshold	a numeric value used to determine zero elements in the loading. This serves mostly to correct approximation errors.
rho	penalty parameter. When method="SPCA", we need further regularizations for the case when the number of variables is larger than the number of observations. We refer to Zou et al. (2006) and Bauer (2022) for more details.
max.iter	maximum number of iterations.
trace	a logical value indicating if the progress is printed.
eps.conv	a numerical value as convergence criterion.
orthogonal	a logical value indicating if the sparse loadings are orthogonalized.
check	a character string indicating if only rows or rows as well as columns are used to detect the underlying block structure. rows checks if the rows fulfill the required structure. rnc checks if rows and columns fulfill the required structure.
...	further arguments passed to or from other methods.

Value

single or list of pla class containing the following attributes:

x	a numeric matrix or data frame which equals the input of x.
EC	a numeric vector that contains the weight-corrected evaluation criterion (CEC) if criterion="corrected" and the evaluation criterion (EC) if criterion="normal".
loadings	a matrix of variable loadings (i.e. a matrix containing the sparse loadings).
blocks	a list of blocks which are identified by sparse principal loading analysis.
W	a matrix of variable loadings used to calculate the evaluation criterion. If criterion="corrected", W contains an orthogonal matrix with equal weights in the first column of each loading-block. If criterion="normal", W are the loadings.

References

Bauer JO (2022). "Variable selection and covariance structure identification using sparse principal loading analysis." *Working Paper*. Witten DM, Tibshirani R, Hastie TA (2009). "A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis." *Biostatistics*, **10**(3), 515-534. doi:10.1093/biostatistics/kxp008. Zou H, Hastie T, Tibshirani R (2006). "Sparse Principal Component Analysis." *Journal of Computational and Graphical Statistics*, **15**(2), 265–286. ISSN 1061-8600, doi:10.1198/106186006X113430.

Examples

```
#############
## First example: we apply SPLA to a classic example from PCA
#############

spla(USArrests, method = "spca", para=c(0.5, 0.5, 0.5, 0.5), cor=TRUE)

## we obtain two blocks:
## 1x1 (Urbanpop) and 3x3 (Murder, Aussault, Rape).
## The large CEC of 0.922 indicates that the given structure is reasonable.

spla(USArrests, method = "spca", para=c(0.5, 0.5, 0.7, 0.5), cor=TRUE)

## we obtain three blocks:
## 1x1 (Urbanpop), 1x1 (Rape) and 2x2 (Murder, Aussault).
## The mid-ish CEC of 0.571 for (Murder, Aussault) indicates that the found
## structure might not be adequate.

#############
## Second example: we replicate a synthetic example similar to Bauer (2022)
#############

set.seed(1)
N = 500
V1 = rnorm(N,0,10)
V2 = rnorm(N,0,11)

## Create the blocks (X_1,...,X_4) and (X_5,...,X_8) synthetically

X1 = V1 + rnorm(N,0,1) #X_j = V_1 + N(0,1) for j =1,...,4
X2 = V1 + rnorm(N,0,1)
X3 = V1 + rnorm(N,0,1)
X4 = V1 + rnorm(N,0,1)

X5 = V2 + rnorm(N,0,1) #X_j = V_1 + N(0,1) for j =5,...9
X6 = V2 + rnorm(N,0,1)
X7 = V2 + rnorm(N,0,1)
X8 = V2 + rnorm(N,0,1)

X = cbind(X1, X2, X3, X4, X5, X6, X7, X8)

## Conduct SPLA to obtain the blocks (X_1,...,X_4) and (X_5,...,X_8)
```

```
## use method = "pmd" (default)
spla(X, para = 1.4)

## use method = "spca"
spla(X, method = "spca", para = c(500,60,3,8,5,7,13,4))
```

`str,Block-method` *Block - str*

Description

Generic function to create a string out of the blocks structure.

Usage

```
## S4 method for signature 'Block'
str(object)
```

Arguments

`object`	block.

Value

A string representing the Block.

Examples

```
block <- new("Block", features=c(2, 5), explained_variance=0.03)
str(block)
```

Index