spca package help files

Giovanni Merola

December 11, 2014

spca-package

`spca-package`	*Utilities for computing Sparse Principal Components with the LS SPCA method.*

Description

Sparse principal components have few loadings different from zero. The functions in this package compute the sparse components with the method LS SPCA. These solutions attain the Least Squares approximation to the data using a correlation (or covariance) matrix.

The solutions are obtained either through a Branch-and-Bound search (`spcabb`) or a more efficient iterative backward Elimination Algorithm (`spcabe`).

If the indices of the sparse loadings are known, the LS SPCA solutions can be computed with `spca`

The output is an object of class spca. The minimal spca object contains the following elements:

loadings	A matrix with the loadings scaled to unit L_2 norm in the columns.
vexp	A vector with the % variance explained by each component.
vexpv	A vector with the % variance explained by each principal component.
ind	A list of the indices of the sparse loadings.

The following methods are available

`print.spca`	Prints the nonzero loadings
`plot.spca`	Plots the variance explained and the nonzero loadings
`summary.spca`	Prints summary statistics of the solutions
`showload`	shows and plots the spca loadings. Not implemented as an spca method.
`compare.spca`	Compares different spca objects, giving summaries and plots. Not implemented as an spca me

References

Giovanni M. Merola. 2014. *Least Squares Sparse Principal Component Analysis: a Backward Elimination approach to attain large loadings.* To appear on Austr.&NZ Jou. Stats. Giovanni M. Merola.

Giovanni M. Merola. 2014. *Sparse Principal Component Analysis: a Least Squares approximation approach.* `http://arxiv.org/abs/1406.1381`

See Also

`spcabb` and `spcabe for usage examples`.

anthrop

`anthrop`	*Anthropometric measures of criminals*

Description

This dataset was used for the first application of PCA. It consists of the correlation matrix of seven measures of physical characteristics of a random sample of British criminals. This dataset was used for the first PCA application (by hand!). Useful for testing.

Format

A 7 by 7 correlation matrix.

`Head Length`

`Head Breadth`

`Face Breadth`

`Finger`

`Cubit`

`Foot`

`Height`

References

Macdonell, W. (1902). Criminal Anthropometry and the Identification of Criminals. *Biometrika*, 1(2):177-227.

bsbl

bsbl	*Baseball hitters career and 1986 season total statistics*

Description

Correlation matrix of 16 statistics of major league hitters some of the overall career and others relative to the 1986 season. Available at StatLib. The matrix has a block structure, defined by season offensive play, career offensive play and season defensive play.

Format

A *16* by *16* correlation matrix.

`TAB_86`	times at bat in 1986
`HIT_86`	hits in 1986
`HR_86`	home runs in 1986
`RUN_86`	runs in 1986
`RB_86`	runs batted-in in 1986
`WAL_86`	walks in 1986
`YC`	years in the major leagues
`TAB`	times at bat during his career
`HIT`	hits during his career
`HR`	home runs during his career
`RUN`	runs during his career
`RUNB`	runs batted-in during his career
`WAL`	walks during his career
`PO_86`	put outs in 1986
`ASS_86`	assists in 1986
`ERR_86`	errors in 1986

Source

`http://lib.stat.cmu.edu/datasets/baseball.data`

bsbl_avg

`bsbl_avg` *Baseball hitters career and 1986 season average statistics*

Description

Same data as above after averaging the career totals with the years in career.
The matrix no longer has a block structure.

Format

A *16* by *16* correlation matrix. See `bsbl` for variables names.

Source

`http://lib.stat.cmu.edu/datasets/baseball.data`

bsbl_labels

`bsbl_labels` *Baseball hitters statistics labels reference table*

Description

This data frame provides descriptive labels for the variables in the bsbl datasets matching the short ones used.

Format

A *16* by *16* correlation matrix.

Source

`http://lib.stat.cmu.edu/datasets/baseball.data`

choosecard

choosecard	*Function for choosing the cardinality of the sparse components*

Description

Interactive function that produces and plots various statistics relative to different cardinalities of the sparse components.

Usage

```
choosecard(S, method = c("BE", "BB"), perc = TRUE, unc = TRUE, trim = 1,
  reducetrim = TRUE, prntrace = FALSE, cardstoprint, interact = TRUE,
  rtntrace = TRUE, doplot = TRUE, plotminload = TRUE,
  plotcvexp = c("rel", "abs", FALSE), plotlovsvexp = TRUE,
  plotentropy = TRUE, plotfarcomeni = FALSE, mfrowplot = 2,
  mfcolplot = 2, cardstoplot, ce = 1)
```

Arguments

`S`	A correlation (or covariance) matrix.
`method`	String. Method used to produce solutions, either BE or BB.
`perc`	Logical: should the loading be scaled as percentages?
`unc`	Logical vector. If TRUE the corresponding component is computed uncorrelated, otherwise correlated. Can be shorter than nd. See details for spcabe.
`trim`	Number of loadings to trim at each iteration. See details for spcabe
`reducetrim`	Logical. If TRUE and trim ¿ 1 when are left less than trim + mincard[j] loadings, trim is reduced to 1 for these last loadings.
`prntrace`	Logical: should the trace of the trimming be printed?
`cardstoprint`	Integer: number of cardinalities to print with the trace. If missing all, otherwise only the last *cardstoprint* solutions will be printed.
`interact`	Hybrid: if TRUE the cardinalities chosen must be entered intearctively. If a vector of cardinalities is passed the plots and tracies are produced.
`rtntrace`	Logical: should the trace of the trimming be returned?
`doplot`	Logical: should the any plotting be done
`plotminload`	Logical: should the minimum loading (or contribution if perc = TRUE) for each cardinality be plotted?
`plotcvexp`	string: if set =*abs* the percentage variance explained for each cardinality is plotted. If set =*rel* the percentage cumulative variance explained relative to that explained by the same number of PCs is plotted against the cardinality. If set =FALSE (or anything else) none is plotted.
`plotlovsvexp`	Logical: should the variance explained for each cardinality be plotted against the minimal loadings?
`plotentropy`	Logical: should the entropy of the loadings be plotted?
`plotfarcomeni`	Logical: should the sparsity index proposed by farcomeni be plotted?
`mfrowplot,mfcolplot`	Integers. The number of rows and columns on which display the plots
`cardstoplot`	Integer: number of cardinalities to plot. If missing all are plotted
`ce`	Real ¿ 0. The expansion factor for the plots labels.

Details

This function is interactive produces plots relative to the different cardinalities of a component and then asks which cardinality is preferred and computes the next. The process can be stopped by entering adding a decimal value to cardinality of the last component desired. By default the solutions are computed with the BE algorithm (`spcabe`). *prntrace=TRUE* prints the trace of the last *cardstoprint* trimmings, with the variables orderd in elimination order during cardinality selection. The order may not be univocal when using the BB algorithm. *rtntrace* returns the full trace for all dimensions. The default settings produce 4 plots. Farcomeni's index is computed as $vexp(c_j) - \frac{\log(c_j)\bar{\sigma}^2}{j+1}$, where c_j is the cardinality and $\bar{\sigma}^2$ is the average variance (=1 for correlation matrices). The values of farcomeni's Index and entropy are not returned if their plot is not required.

Value

If rtntrace = TRUE a list of matrices of full traces is retuned.

Note

The plots are not very customisable. Personalised plots can be easily produced from the spca object.

References

Giovanni M. Merola. 2014. Least Squares Sparse Principal Component Analysis: a backward elimination approach to attain large loadings. To appear in Australian and New Zealand Journal of Statistics.

See Also

`spcabe`, `spcabb`.

Examples

```
## Not run:
data(bsbl)
## run choosecard in non interactive mode
ba <- choosecard(bsbl, prntrace = TRUE, cardstoprint = 6, doplot = FALSE, interact = c(3, 3, 4))

# to run in interactive mode replace the interact and doplot arguments with TRUE
# (or remove them from the call altogether).

## End(Not run)
```

compare

compare.spca *Compares two or more spca solutions*

Description

Compares two or more spca solutions by printing the loadings and the summary statistics next to each other. It can plot the cumulative variances explained together with those of PCA and the loadings for each component.

Usage

```
## S3 method for class 'spca'
compare(smpc, compareto, nd, methodsnames, perc = TRUE,
  plotvar = TRUE, plotload = FALSE, labelload = TRUE,
  sizelabelsload = 0.85, poslabeload = 3, prnload = TRUE,
  shortnamescomp = TRUE, rtn = FALSE, prn = TRUE, only.nonzero = TRUE,
  bnw = FALSE, mfrowload = 1, mfcolload = 1, sizelegend = 0.85, ...)

compare(smpc, ...)
```

Arguments

`smpc`	An spca object
`compareto`	A list of spca objects with which smpc is to be compared. Can be givenas single object
`nd`	Number of dimensions to compare. If not specified set to the minimum number of loadings in the objects.
`methodsnames`	Names for each object included. If not specified, labels are created as Met1, Met2, etc.
`perc`	Logical: should the loadings be standardised to unit L_1 norm (and printed as percentage contributions).
`plotvar`	Logical: should the cumulative variances be plotted?
`plotload`	Logical or integer (¿0): should the loadings be plotted and how many?
`labelload`	Logical: write variables names loading plots?
`sizelabelsload`	Real: expansion coefficient for loading plot label. See *cex* in `par`,
`poslabeload`	integer: position of the labels of the laodings. 1 = bottom, 2 = left, 3 = top (default), 4 = right.
`prnload`	Logical or Integer (¿0): should the loadings be printed and how many?
`shortnamescomp`	Logical: should the loadings be printed with short names (Cx.y) or long ones (Cx.methodsnames)?
`rtn`	Logical: should the text table of loadings and the matrix of summaries be returneded?
`prn`	Logical: should anything be printed? Takes priority on prnload.
`only.nonzero`	Logical: should only nonzero contributions be printed?
`bnw`	Logical: should plots be in blck and white?

`mfrowload`	Number of loadings plots per row.
`mfcolload`	Number of loadings plots per column.
`sizelegend`	Magnification of the legend labels, see *cex* in `par`.
`...`	additional arguments for generic compare. Disabled, additional arguments will generate an error.

Details

For the meaning of each summary statistic see `summary.spca`. Plotvar plots *nd* values. if *plotload* or *prnload* are integer, that number of loaidngs will be processed. However, *nd* loadings are always returned if *rtn=TRUE*.

Value

If rtn = TRUE, it returns a formatted text table with the loadings and a matrix with the summaries.

`NULL`

See Also

Examples in `spcabb` and `spcabe`.

compare.spca

`compare.spca`	*Compares two or more spca solutions*

Description

Compares two or more spca solutions by printing the loadings and the summary statistics next to each other. It can plot the cumulative variances explained together with those of PCA and the loadings for each component.

Usage

```
## S3 method for class 'spca'
compare(smpc, compareto, nd, methodsnames, perc = TRUE,
  plotvar = TRUE, plotload = FALSE, labelload = TRUE,
  sizelabelsload = 0.85, poslabeload = 3, prnload = TRUE,
  shortnamescomp = TRUE, rtn = FALSE, prn = TRUE, only.nonzero = TRUE,
  bnw = FALSE, mfrowload = 1, mfcolload = 1, sizelegend = 0.85, ...)

compare(smpc, ...)
```

Arguments

`smpc`	An spca object
`compareto`	A list of spca objects with which smpc is to be compared. Can be givenas single object
`nd`	Number of dimensions to compare. If not specified set to the minimum number of loadings in the objects.
`methodsnames`	Names for each object included. If not specified, labels are created as Met1, Met2, etc.
`perc`	Logical: should the loadings be standardised to unit L_1 norm (and printed as percentage contributions).
`plotvar`	Logical: should the cumulative variances be plotted?
`plotload`	Logical or integer (¿0): should the loadings be plotted and how many?
`labelload`	Logical: write variables names loading plots?
`sizelabelsload`	Real: expansion coefficient for loading plot label. See *cex* in `par`,
`poslabeload`	integer: position of the labels of the laodings. 1 = bottom, 2 = left, 3 = top (default), 4 = right.
`prnload`	Logical or Integer (¿0): should the loadings be printed and how many?
`shortnamescomp`	Logical: should the loadings be printed with short names (Cx.y) or long ones (Cx.methodsnames)?
`rtn`	Logical: should the text table of loadings and the matrix of summaries be returneded?
`prn`	Logical: should anything be printed? Takes priority on prnload.
`only.nonzero`	Logical: should only nonzero contributions be printed?
`bnw`	Logical: should plots be in blck and white?
`mfrowload`	Number of loadings plots per row.
`mfcolload`	Number of loadings plots per column.
`sizelegend`	Magnification of the legend labels, see *cex* in `par`.
`...`	additional arguments for generic compare. Disabled, additional arguments will generate an error.

Details

For the meaning of each summary statistic see `summary.spca`. Plotvar plots *nd* values. if *plotload* or *prnload* are integer, that number of loaidngs will be processed. However, *nd* loadings are always returned if *rtn=TRUE*.

Value

If rtn = TRUE, it returns a formatted text table with the loadings and a matrix with the summaries.

`NULL`

See Also

Examples in `spcabb` and `spcabe`.

is.spca

`is.spca` *Verifies if an object is of class spca*

Description

Verifies if an object is of class spca

Usage

```
is.spca(x)
```

Arguments

`x`	Any object suspected of being of class spca.

Value

Logical: TRUE if object is of class spca, FALSE otherwise.

pca

`pca` *Computes principal components solutions*

Description

Computes PCA components loadings.

Usage

```
pca(S, nd, only.values = FALSE, screeplot = FALSE, kaiser.print = FALSE)
```

Arguments

`S`	A correlation or covariance matrix.
`nd`	Integer: number of loadings to retain. If missing all loadings are retained.
`only.values`	Logical: should only the eigenvalues be computed?
`screeplot`	Logical: should the screeplot be plotted?
`kaiser.print`	Logical: should the kaiser rule be computed, printed and returned?.

Details

nd is just the number of components retained from the full eigen decomposition, doesn't speed up the function. *only.values* does not compute the loadings and is more efficient. Kaiser rule determines the number of components as the number of eigenvalues larger than one. It should be used only for correlation matrices, if called on a covariance matrix a warning is generated.

Value

An object of class *spca* is returned, which contains:

`loadings`	The matrix of loadings (if only.values = TRUE it is equal to NULL).
`vexpv`	a vector of variances explained by each PC
`vexp`	a vector of variances explained by each PC
`cvexp`	a vector of cumulative variances explained by the PCs

In addition, if `kaiser.print = TRUE`:

`kaiser`	The number of eigenvalues larger than one.

See Also

See also `print.spca, summary.spca`

Examples

```
## Not run:
 data(anthrop, package = "spca")
 # computes 4 PCs loadings plotting the screeplot and printing the kaiser rule
 mypca <- pca(anthrop, nd = 4, screeplot = TRUE, kaiser.print = TRUE)
 ## print loadings
 mypca
 summary(mypca)

## End(Not run)
```

plot.spca

`plot.spca` *Plots loadings and variance explained for an spca object*

Description

Plots coefficients and variance explained for spca solutions.

Usage

```
## S3 method for class 'spca'
plot(x, cols, plotvexp = TRUE, methodname = FALSE,
  plotload = FALSE, thresh = 0.001, perc = TRUE, variablesnames = FALSE,
  onlynonzero = TRUE, plotloadvsPC = FALSE, pcs = NULL,
  addlabels = TRUE, mfrowload = 1, mfcolload = 1, bnw = FALSE,
  rotlabels = 0, sizelabels = 1, ...)
```

Arguments

`x`	An spca object.
`cols`	The number of components to be plotted. Default all. If an iteger is passed, it is set to 1:cols.
`plotvexp`	Logical: should the cumulative variance explained be plotted?
`methodname`	Name of the method. If FALSE set to LS SPCA
`plotload`	Logical: should the loadings be plotted?
`thresh`	Real: value below this are considered zero and not plotted. It *thresh¿* 0.001 it is effective regardless of the value of *onlynonzero*.
`perc`	Logical: should the loading be scaled as percentages?
`variablesnames`	names of the variables to use in plot of loadings. If FALSE, names are set to V1, V2,... If TRUE the rownames of the matrix of loadings are used.
`onlynonzero`	Logical: should only the non-zero loadings be plotted?
`plotloadvsPC`	Logical: if TRUE the sparse loadings are plotted versus the corresponding PCA ones.
`pcs`	An spca object containing the PCA loadings, typically obtained with the function pca.
`addlabels`	Hybrid: if TRUE the nonzero loadings in the plotloadvsPC and plotload plots are labelled with short names V1, V2,..., if equal to "orig" the original variables names are used as labels, if FALSE no labels are added.
`mfrowload`	Number of loadings plots per row.
`mfcolload`	Number of loadings plots per column.
`bnw`	Logical: should the plots be in black and white?
`rotlabels`	Angle for the rotation of the labels, see *srt* in `par`.
`sizelabels`	Magnification of the labels, see *cex* in `par`.
`...`	Additonal arguments for generic plot. Disabled, additional arguments will generate an error.

Details

The cumulative variance explained is always plotted together with that explained by the PCs. The loadings are plotted as barplots. For large matrices it is reccommended to set onlynonzero = TRUE and variablesnames = F. The plots of the sparse loadings versus the PC's ones are marked with the line of equality of the PCs ones.

Value

None

Note

The value of *thresh* must be chosen according to the value of *perc*.
The "dots" are disabled so that only exact (or partial) prescribed arguments can be entered.
The plots are not very customisable. Personalised plots can be easily produced from the spca object.

See Also

Examples in `spcabe` and `spcabb`. For plotting two or more spca solutions together see `compare`.

print.spca

`print.spca`	*Prints the sparse loadings from an spca object*

Description

Prints sparse loadings omitting the zero ones and giving the cumulative variance explained.

Usage

```
## S3 method for class 'spca'
print(x, cols, only.nonzero = TRUE, perc = TRUE,
  digits = 3, thresh = 0.001, rtn = FALSE, namescomp = NULL, ...)
```

Arguments

`x`	An spca object.
`cols`	A vector indicating which components should be printed. Default all. If an iteger is passed, it is set to 1:cols.
`only.nonzero`	Logical: if = TRUE only the nonzero loadings are printed. otherwise all loadings are printed.
`perc`	Logical: should the loadings be standardised to unit L_1 norm (and printed as percentage contributions)?
`digits`	Integer: number of decimal figures.
`thresh`	Value below which loadings are considered zero and not printed.
`rtn`	Logical: should the formatted (text) table be returned?
`namescomp`	A vector of names for the components. If NULL assigned as "Comp j"
`...`	Additonal arguments for generic print, additional arguments will generate an error.

Value

If rtn = TRUE, it returns a text table formatted as specified by the arguments.

Note

This is a wrapper for the main function in which the "dots" are disabled so that only exact (or partial) prescribed arguments can be entered.

See Also

Examples in spcabb and spcabe.

showload

`showload` *Shows the sparse loadings*

Description

Shows the non-zero loadings separately for each component.

Usage

```
showload(smpc, cols, perc = TRUE, digits = 3, variablesnames = FALSE,
  thresh = 0.001, rtn = FALSE)
```

Arguments

`smpc`	A list of spca objects, typically from spcabe and spcabb. It can also be a simple matrix of loadings.
`cols`	A vector containg the indices of the loadings to be shown. Can be a single value. if missing all loadings are shown: If an integer is passed, only that dimension will be returned.
`perc`	Logical: should the loodings be standardised to unit L_1 norm (and printed as percentage contributions).
`digits`	Number of decimal digits to show.
`variablesnames`	Hybrid: if not FALSE, need to pass a vector of varaiable names.
`thresh`	Loadings with absolute value below this are considered zero.
`rtn`	Logical: should the text table of loadings and the matrix of summaries be returneded?

Details

Useful for large matrices to see the loadings at the same time or to assign long descriptive names.

variablesnames must have the names of the p variables in the first p positions.

Value

If rtn = TRUE, it returns a list with the loadings.

See Also

print.spca, plot.spca. Examples in `spcabe`

spca

`spca` *Computes sparse principal components solutions*

Description

Computes LS SPCA sparse principal components loadings for a given set of indices. See the package vignettes for details.

Usage

```
spca(S, ind, unc = TRUE)
```

Arguments

`S`	A correlation or covariance matrix.
`ind`	A list of indices for each dimension. The number of dimensions to compute is determined by its length. If only the first dimension is required, it can be a vector.
`unc`	A logical vector indicating which components should be should be computed uncorrelated to the preceeding ones. Can be shorter than the number of dimensions to compute. See details.

Details

The number of components to compute is determind from the length of *ind*. If *unc* has fewer elements than the number of indices passed, the remaining elements are set equal to the last one.

Value

An object of class *spca* is returned. It is the smallest instance of an spca object, which contains:

`loadings`	The matrix of loadings
`contributions`	Matrix of loadings scaled to unit L_1 norm.
`vexpv`	a vector of variances explained by each component
`vexp`	a vector of variances explained by each PC

In addition, if `any unc[j] = FALSE`:

`corComp`	The matrix with correlations among components.
`loadingsUnc`	Loadings of the components made uncorrelated.

See Also

`spcabb`, `spcabe`, `summary.spca`

Examples

```
## Not run:
 data(anthrop, package = "spca")
 # for uncorrelated components
 myspca <- spca(anthrop, ind = list(1:2, 3:7))
 ## print loadings
 myspca
 ## print summaries
 summary(myspca)
 # for correlated components
 myspcac <- spca(anthrop, ind = list(1:2, 3:7), unc = FALSE)
 myspcac
 summary(myspcac)
 ## print correlation between components
 myspcac$corComp
 ## print loadings of components made uncorrelated
 myspcac$loadingsUnc
 ## compare the two results numerically and graphically
compare(myspca, myspcac, methodsnames = c("Unc", "Cor"), shortnamescomp = FALSE)

## End(Not run)
```

spcabb

spcabb	*SPCA by Branch-and-Bound*

Description

Finds the LS SPCA loadings with given cardinalities using Branch-and-Bound

Usage

```
spcabb(S, card, unc = TRUE, startind, excludeload = FALSE, nvexp = FALSE,
  msg = TRUE)
```

Arguments

S	A correlation or covariance matrix.
card	A vector of cardinalities for each component. the number of dimensions to compute is determined from its length.
unc	A logical vector indicating if each component should be should be uncorrelated to the preceeding ones or not.
startind	A list of indices from which the sparse loadings will be computed. If missing all combinations of indices are searched.
excludeload	Logical: vector (length nd or shorter) should indices of non-zero loadings in previous components be excluded from future searches?
nvexp	Logical. If TRUE the real variance estimated is used as objective function. Otherwise an approximated form of the variance explainedis used.
msg	Logical: should messages be printed after each component is computed

Details

If unc = FALSE, when nvexp = TRUE the objective function is the true variance explained, otherwise the approximated one (see references or Vignettes for details). unc and excludeload can be vectors of length less than nd (hence also a single value), in this case, the last element is assigned to the missing ones. The BB search is computationally demanding, for large problems (n ¿ 50) consider using `spcabe` Just in case the functions takes too long and it is interrupted, a minimal output of the loadings computed is returned. This is a minimal spca object with elements A, the loadings so far computed, vexp and vexpv.

Value

An object of class *spca* is returned. It contains:

`loadings`	The matrix of loadings
`vexp`	A vector of variances explained by each component
`vexpPC`	A vector of variances explained by each PC
`ind`	A list containing the indices of the non-sparse loadings
`niter`	Number of iterations to compute each component

Call arguments

`unc`	the argument unc passed
`nvexp`	The argument nvexp passed

If `any unc[j] = FALSE`

`corComp`	Matrix with correlations among components.
`Aunc`	Loadings of components made uncorrelated.

Note

Thanks to Dr Alessio Farcomeni for making avilable his R code for the BB algorithm. This version is a slight variation of it.

Author(s)

Giovanni Merola

See Also

`spcabe`

Examples

```
## Not run:
data(anthrop)
## 3 uncorrelated components with sparse loadings each of cardinality
## 3, 3 and 2 (the last value will be set to 3 because uncorrelated components
## must have cardinality at least equal to their order)
myspca1 <- spcabb(anthrop, card = c(3,3, 2) )
# print the results
myspca1
# print summary results
summary(myspca1)
```

```
# show how many iterations each compnent took
myspca1$niter
# plot loadings and cumulative variance explained (4 plots)
plot(myspca1, plotload = TRUE, onlynonzero = FALSE, variablesnames = TRUE)

## 3 correlated components with sparse loadings also each of cardinality 3
myspca2 <- spcabb(anthrop, card = rep(3, 3), unc = FALSE, nvexp = FALSE )
# print results
myspca2
# print summary results
summary(myspca2)
# show how many iterations each compnent took
myspca2$niter
## compare the correlated with the uncorrelated solutions
compare(myspca1, myspca2, methodsnames = c("unc", "cor"))
# print the correlations among the correlated components
myspca2$corComp
# print the loadings of components made uncorrelated
myspca2$loadingsUnc

## 3 correlated components with sparse loadings of cardinality 2, 1 and 3
myspca3 <- spcabb(anthrop, card = c(2, 1, 1), unc = FALSE )
# print the results
myspca3
# print summary results
summary(myspca3)
## print correlation between components
myspca3$corComp
## print loadings of components made uncorrelated
myspca3$loadingsUnc

## End(Not run)
```

spcabe

spcabe	*SPCA by Backward Elimination algorithm*

Description

Computes LS SPCA components by iteratively trimming small loadings.

Usage

```
spcabe(S, nd = FALSE, ndbyvexp = FALSE, mincard = NULL, thresh = FALSE,
  threshvar = FALSE, threshvaronPC = FALSE, perc = TRUE, unc = TRUE,
  trim = 1, reducetrim = TRUE, startind = NULL, excludeload = FALSE,
  diag = FALSE, choosecard = NULL, eps = 1e-04, msg = TRUE)
```

Arguments

S	A correlation or covariance matrix.

`nd`	Integer. Number of dimensions to compute. If FALSE and ndbyvexp ¡ 1 the number of components is determined by the latter value. If FALSE and ndbyvexp = 1 or = FALSE the program will give an error.
`ndbyvexp`	Real in [0,1] or FALSE. Minimum percentage of total variance explained by the components computed. If reached before the specified nd, it takes priority.
`mincard`	Vector of minimal cardinality of each components. If FALSE and unc[j] = TRUE, the j-th value is set to j, otherwise all values are set to 1. Takes priority on other controls on trimming.
`thresh`	Vector of values below which loadings are trimmed. Can be shorter than nd. See details.
`threshvar`	Vector of minimal percentage of variance loss from the full initial solution allowed for each component. If reached current trimming is cancelled and solution returned. If FALSE it is set to 1.
`threshvaronPC`	Vector of minimal total percentage of variance loss from the total variance explained by the PCs allowed to trimming. If reached current trimming is cancelled and solution returned. If FALSE it is set to 1. It takes priority over threshvar[j], if both specified.
`perc`	Logical: does the threshold refers to the percentage contributions (the loadings scaled to unitary L1 norm)?
`unc`	Logical vector. If TRUE the corresponding component is computed uncorrelated, otherwise correlated. Can be shorter than nd. See details.
`trim`	Number of loadings to trim at each iteration. mincard[j] takes priority if conflicting.
`reducetrim`	Logical. If TRUE and trim ¿ 1 when are left less than trim + mincard[j] loadings, trim is reduced to 1 for these last loadings.
`startind`	List of vectors with the initial set of indices for each component. If NULL, the full set of indices (1:ncol(S)) is assigned to each component
`excludeload`	Logical: vector (length nd or shorter) should the indices of non-zero loadings in previous components be excluded from future searches?
`diag`	Logical: should diagnostic output be returned?.
`choosecard`	NULL or Integer. Setting the value to an integer makes the function return a full trace of the elimination for that component. It is used by the choosecard function.
`eps`	Value below which the absolute value of a loading is considered zero.
`msg`	Logical: should messages be printed after each component is computed

Details

Sparse loadings are computed by iteratively trimming the ones smaller than thresh[j] for each component. If *ndbyvexp* ¡ 1, the algorithm will stop when that percentage of total Vexp is reached with the last component computed.

Arguments *threshvar*, *threshvaronPC*, *thresh*, *excludeload* and *unc* can be entered with fewer elements than the number of components to compute, *nd*. In this case, or if *nd* is determined by the variance explained, the missing elements are set equal to the last one entered (also if just one value is given). The same is true for *mincard* but for the components required to be uncorrelated their values are set equal to the order of the component.

startind can be set for the first few components, the following will be computed on the whole set of variables.

Trimming stops if mincard[j] is reached. Trimming is controlled in two more optional ways: if the last trimming caused a loss of variance explained from the initial solution greater than *threshvar* or the loss of proportion of total variance explained over the corresponding PCA value drops below the specified percentage *threshvaronPC*. The rules can be used together, setting the values to FALSE or 1 to avoid them.

When *excludeload* = TRUE or *startindex* is set, the cardinality of the starting indices could be less than the order of the component to compute. In this case uncorrelatedness cannot be achieved and the component will be computed as correlated. The flag *converged* will be set to 3 and a warning message printed.

If vector arguments of length less than the number of components to compute are passed (hence also if a single one is passed), the last element is assigned to the missing ones.

Value

spcabe returns an object of class *spca*. On top of the basic elements of spca objects, it contains other ones useful for diagnostics and analysis. Some elements are present only if some of the arguments are activated. The object contains the following components:

`loadings`	Matrix with the loadings scaled to unit L_2 norm in the columns.

If `perc = TRUE`

`contributions`	Matrix of loadings scaled to unit L_1 norm.
`vexp`	Vector with the % variance explained by each component.
`vexpPC`	Vector with the % variance explained by each principal component.
`cardinality`	Vector with the cardinalities of each loadings.
`ind`	List with the indices of the non-zero loadings for each component.
`unc`	the argument unc passed.
`converged`	Vector with the stop for trimming: 0 by *thresh*, 1 by *mincard*, 2 by *threshvar* or *threshvaronPC*. The value 3 means that uncorrelatedness could not be achieved because too few indices were available (see notes).

If any `unc[j] = TRUE`

`corComp`	Matrix of correlations among the sparse components
`Aunc`	Loadings of components made uncorrelated

If `diag == TRUE` a number of details are returned:

`vexpo`	Vector with the % variance explained by the initial untrimmed components.
`totvcloss`	Vector with the % loss in total variance explained including each component over that explained by the corresponding PC (vexpPC - vexp)/vexpPC.
`vlossbe`	Vector with the % loss in variance explained loss by trimming over that explained by the initial component (vexpo).
`niter`	Vector with number of iterations for each trimming round.
`eliminated`	List of indices of loadings eliminated for each component

Call arguments, possibly modified by the algorithm:

`thresh`	Vector of tresholds for the size of loadings
`threshvar`	Vector of tresholds on loss of variance explained by each component
`ndbyvexp`	Required total variance explained
`stopbyvar`	Logical, did the algorithm terminate because the required total variance explained was reached?
`mincard`	Minimal cardinalities required

See Also

`spcabb`, `summary.spca`, `compare.spca`.

Examples

```
## Not run:
  "Note the warnings and messages produced by the examples"
  data(anthrop, package = "spca")

  # 3 basic spcabe components with default values,
  # since uncorrelated component these have card = 1, 2, and 3
  myspca1 <- spcabe(anthrop, nd = 3)
  myspca1
  summary(myspca1)
  ## plot the results
  plot(myspca1, plotload = TRUE, onlynonzero = FALSE, mfrowload = 3, variablesnames = TRUE)

  ## spcabe with 3 components trimmed to different thresholds and mincard
  myspca2 <- spcabe(anthrop, nd = 3, thresh = c(0.3, 0.25, 0.15), mincard = c(2,3,3))
  summary(myspca2)
  myspca2
  # show the first two loadings as percentage contributions
  showload(myspca2, cols = 1:2, perc = TRUE)

  ## spcabe requirig explaining at least 75% of total variance and that each component
  ## explains at least 95% of variance explained by the pcs (see details)
  myspca3 <- spcabe( anthrop, ndbyvexp = 0.75, threshvaronPC = 0.95)
  summary(myspca3)
  myspca3
  # compare the three solutions
  compare(smpc = myspca1, compareto = list( myspca2, myspca3),
  methodsnames = c("myspca1", "myspca2", "myspca3"))

## End(Not run)
```

summary.spca

`summary.spca` *Prints summaries from an spca object*

Description

Prints summaries and comparisons with the full PCA solutions for a set of LS SPCA loadings.

Usage

```
## S3 method for class 'spca'
summary(object, cols, perc = TRUE, rtn = FALSE, prn = TRUE,
  thrsehcard = 0.001, ...)
```

Arguments

`object`	An spca object.
`cols`	A vector indicating which components should be included. Default all. If an iteger is passed, it is set to 1:cols.
`perc`	Logical: should the laodings be standardised to unit L1 norm (and printed as percentage contributions)
`rtn`	Logical: should the summary matrix of summaries be returneded?
`prn`	Logical: should anything be printed? Takes priority on prnload.
`thrsehcard`	Value below which loadings are considered zero and not counted in the cardinality
`...`	Additonal arguments for generic summary, additional arguments will generate an error.

Details

The summaries are printed as formatted text, if rtn = TRUE, the value returned is a numerical matrix.

For each component the following summaries are computed:

PVE	The percentage variance explained
PCVE	The percentage cumulative variance explained
PRCVE	The percentage cumulative variance explained relative to that of the corresponding principal components
Card	The cardinality, that is the number of non zero loadings
Ccard	The cumulative cardinality.
PVE/Card	The percentage variance explained over the cardinality.
PCVE/Ccard	The percentage cumulative variance explained over the cumulative cardinality.
Converged	If the object was computed with *spcabe*, type of convergence: 0 if all loadings bigger than *thresh*, 1 if minimal cardinality reached or 2 if the maximal variance loss in trimming was reached.
MinLoad	Minimum absolute value of the non-zero loadings.

If perc = TRUE, the last row gives the minimum absolute percentage contribution, MinPContr

Value

If rtn = TRUE, a numerical matrix with the summaries.

Note

This is a wrapper for the main function in which the ”dots” are disabled so that only exact (or partial) prescribed arguments can be entered.

See Also

Examples in spcabe and spcabb