Package 'rospca'

July 23, 2025

Version 1.1.1

Date 2024-11-30

Title Robust Sparse PCA using the ROSPCA Algorithm

Description Implementation of robust sparse PCA using the ROSPCA algorithm of Hubert et al. (2016) <DOI:10.1080/00401706.2015.1093962>.

Maintainer Tom Reynkens <tomreynkens.r@gmail.com>

Depends R (>= 2.14.0)

Imports stats, graphics, parallel, mrfDepth (>= 1.0.5), robustbase (>= 0.92-6), pcaPP, rrcov, elasticnet, mvtnorm, pracma

License GPL (>= 2)

URL https://github.com/TReynkens/rospca

BugReports https://github.com/TReynkens/rospca/issues

ByteCompile yes

NeedsCompilation no

Author Tom Reynkens [aut, cre] (ORCID: <https://orcid.org/0000-0002-5516-5107>), Valentin Todorov [ctb] (Original R code for PcaHubert and diagnostic plot in rrcov package), Mia Hubert [ctb], Eric Schmitt [ctb], Tim Verdonck [ctb]

Repository CRAN

Date/Publication 2024-12-02 09:20:10 UTC

Contents

`angle` *Standardised last principal angle*

Description

Standardised last principal angle between the subspaces generated by the columns of `A` and `B`.

Usage

```
angle(A, B)
```

Arguments

`A`	Numeric matrix of size p by k.
`B`	Numeric matrix of size q by l.

Details

We compute the last principal angle between the subspaces generated by the columns of `A` and `B` using the algorithm in Bjorck and Golub (1973). This angle takes values between 0 and $\pi/2$. We divide it by $\pi/2$ to make it take values between 0 and 1, where 0 indicates that the subspaces are close.

Value

Standardised last principal angle between `A` and `B`.

Author(s)

Tom Reynkens

References

Bjorck, A. and Golub, G. H. (1973), "Numerical Methods for Computing Angles Between Linear Subspaces," *Mathematics of Computation*, 27, 579–594.

Examples

```
tmp <- dataGen(m=1)

P <- eigen(tmp$R)$vectors[,1:2]
PP <- rospca(tmp$data[[1]], k=2)$loadings

angle(P, PP)
```

---

dataGen *Generate sparse data with outliers*

Description

Generate sparse data with outliers using simulation scheme detailed in Hubert et al. (2016).

Usage

```
dataGen(m = 100, n = 100, p = 10, a = c(0.9,0.5,0), bLength = 4, SD = c(10,5,2),
        eps = 0, seed = TRUE)
```

Arguments

m	Number of datasets to generate, default is 100.
n	Number of observations, default is 100.
p	Number of dimensions, default is 10.
a	Numeric vector containing the inner group correlations for each block. The number of useful blocks is thus given by $k = length(a) - 1$ which should be at least 2. By default, the correlations are equal to 0.9, 0.5 and 0, respectively.
bLength	Length of the blocks of useful variables, default is 4.
SD	Numeric vector containing the standard deviations of the blocks of variables, default is `c(10,4,2)`. Note that `SD` and `a` should have the same length.
eps	Proportion of contamination, should be between 0 and 0.5. Default is 0 (no contamination).
seed	Logical indicating if a seed is used when generating the datasets, default is `TRUE`.

Details

Firstly, we generate a correlation matrix such that it has sparse eigenvectors. We design the correlation matrix to have $length(a) = k + 1$ groups of variables with no correlation between variables from different groups. The first k groups consist of `bLength` variables each. The correlation between the different variables of the group is equal to `a[1]` for group 1, ... The (k+1)th group contains the remaining $p - k \times bLength$ variables, which we specify to have correlation `a[k+1]`. Secondly, the correlation matrix `R` is transformed into the covariance matrix $\Sigma = V^{0.5} \cdot R \cdot V^{0.5}$, where $V = diag(SD^2)$.

Thirdly, the n observations are generated from a p-variate normal distribution with mean the p-variate zero-vector and covariance matrix Σ. Standard normally distributed noise terms are also added to each of the p variables to make the sparse structure of the data harder to detect.
Finally, $(100 \times eps)\%$ of the data points are randomly replaced by outliers. These outliers are generated from a p-variate normal distribution as in Croux et al. (2013).
The ith eigenvector of R, for $i = 1, ..., k$, is given by a (sparse) vector with the $(bLength \times (i - 1) + 1)$th till the $(bLength \times i)$th elements equal to $1/\sqrt{bLength}$ and all other elements equal to zero.
See Hubert et al. (2016) for more details.

Value

A list with components:

data	List of length m containing all data matrices.
ind	List of length m containing the numeric vectors with the indices of the contaminated observations.
R	Correlation matrix of the data, a numeric matrix of size p by p.
Sigma	Covariance matrix of the data (Σ), a numeric matrix of size p by p.

Author(s)

Tom Reynkens

References

Hubert, M., Reynkens, T., Schmitt, E. and Verdonck, T. (2016). "Sparse PCA for High-Dimensional Data with Outliers," *Technometrics*, 58, 424–434.

Croux, C., Filzmoser, P., and Fritz, H. (2013), "Robust Sparse Principal Component Analysis," *Technometrics*, 55, 202–214.

Examples

```
X <- dataGen(m=1, n=100, p=10, eps=0.2, bLength=4)$data[[1]]

resR <- robpca(X, k=2, skew=FALSE)
diagPlot(resR)
```

---

diagPlot *Diagnostic plot for PCA*

Description

Make diagnostic plot using the output from robpca or rospca.

Usage

```
diagPlot(res, title = "Robust PCA", col = "black", pch = 16, labelOut = TRUE, id = 3)
```

Arguments

res	A list containing the orthogonal distances (od), the score distances (sd) and their respective cut-offs (cutoff.od and cutoff.sd). Output from robpca or rospca can for example be used.
title	Title of the plot, default is "Robust PCA".
col	Colour of the points in the plot, this can be a single colour for all points or a vector specifying the colour for each point. The default is "black".
pch	Plotting characters or symbol used in the plot, see points for more details. The default is 16 which corresponds to filled circles.
labelOut	Logical indicating if outliers should be labelled on the plot, default is TRUE.
id	Number of OD outliers and number of SD outliers to label on the plot, default is 3.

Details

The diagnostic plot contains the score distances on the x-axis and the orthogonal distances on the y-axis. To detect outliers, cut-offs for both distances are added, see Hubert et al. (2005).

Author(s)

Tom Reynkens, based on R code from Valentin Todorov for the diagnostic plot in **rrcov** (released under GPL-3).

References

Hubert, M., Rousseeuw, P. J., and Vanden Branden, K. (2005), "ROBPCA: A New Approach to Robust Principal Component Analysis," *Technometrics*, 47, 64–79.

Examples

```
X <- dataGen(m=1, n=100, p=10, eps=0.2, bLength=4)$data[[1]]

resR <- robpca(X, k=2, skew=FALSE)
diagPlot(resR)
```

Glass *Glass data*

Description

Glass data of Lemberge et al. (2000) containing Electron Probe X-ray Microanalysis (EPXMA) intensities for different wavelengths of 16–17th century archaeological glass vessels. This dataset was also used in Hubert et al. (2005).

Usage

```
data(Glass)
```

Format

A data frame with 180 observations and 750 variables. These variables correspond to EPXMA intensities for different wavelengths and are indicated by `V1`, `V2`, ..., `V750`.

Source

Lemberge, P., De Raedt, I., Janssens, K. H., Wei, F., and Van Espen, P. J. (2000), "Quantitative Z-Analysis of the 16–17th Century Archaelogical Glass Vessels using PLS Regression of EPXMA and μ-XRF Data," *Journal of Chemometrics*, 14, 751–763.

References

Hubert, M., Rousseeuw, P. J., and Vanden Branden, K. (2005), "ROBPCA: A New Approach to Robust Principal Component Analysis," *Technometrics*, 47, 64–79.

Examples

```
data(Glass)

res <- robpca(Glass, k=4, alpha=0.5)
matplot(res$loadings, type="l", lty=1)
```

robpca *ROBust PCA algorithm*

Description

ROBPCA algorithm of Hubert et al. (2005) including reweighting (Engelen et al., 2005) and possible extension to skewed data (Hubert et al., 2009).

Usage

```
robpca (x, k = 0, kmax = 10, alpha = 0.75, h = NULL, mcd = FALSE,
        ndir = "all", skew = FALSE, ...)
```

Arguments

x	An n by p matrix or data matrix with observations in the rows and variables in the columns.
k	Number of principal components that will be used. When k=0 (default), the number of components is selected using the criterion in Hubert et al. (2005).
kmax	Maximal number of principal components that will be computed, default is 10.
alpha	Robustness parameter, default is 0.75.
h	The number of outliers the algorithm should resist is given by $n - h$. Any value for h between $n/2$ and n may be specified. Default is NULL which uses h=ceiling(alpha*n)+1. Do not specify alpha and h at the same time.
mcd	Logical indicating if the MCD adaptation of ROBPCA may be applied when the number of variables is sufficiently small (see Details). If mcd=FALSE (default), the full ROBPCA algorithm is always applied.
ndir	Number of directions used when computing the outlyingness (or the adjusted outlyingness when skew=TRUE), see outlyingness and adjOutl for more details.
skew	Logical indicating if the version for skewed data (Hubert et al., 2009) is applied, default is FALSE.
...	Other arguments to pass to methods.

Details

This function is based extensively on PcaHubert from **rrcov** and there are two main differences:

The outlyingness measure that is used for non-skewed data (skew=FALSE) is the Stahel-Donoho measure as described in Hubert et al. (2005) which is also used in PcaHubert. The implementation in **mrfDepth** (which is used here) is however much faster than the one in PcaHubert and hence more, or even all, directions can be considered when computing the outlyingness measure.

Moreover, the extension for skewed data of Hubert et al. (2009) (skew=TRUE) is also implemented here, but this is not included in PcaHubert.

For an extensive description of the ROBPCA algorithm we refer to Hubert et al. (2005) and to PcaHubert.

When mcd=TRUE and $n < 5 \times p$, we do not apply the full ROBPCA algorithm. The loadings and eigenvalues are then computed as the eigenvectors and eigenvalues of the MCD estimator applied to the data set after the SVD step.

Value

A list with components:

loadings	Loadings matrix containing the robust loadings (eigenvectors), a numeric matrix of size p by k.
eigenvalues	Numeric vector of length k containing the robust eigenvalues.
scores	Scores matrix (computed as $(X - center) \cdot loadings$), a numeric matrix of size n by k.
center	Numeric vector of length k containing the centre of the data.
k	Number of (chosen) principal components.
H0	Logical vector of size n indicating if an observation is in the initial h-subset.
H1	Logical vector of size n indicating if an observation is kept in the reweighting step.
alpha	The robustness parameter α used throughout the algorithm.
h	The h-parameter used throughout the algorithm.
sd	Numeric vector of size n containing the robust score distances within the robust PCA subspace.
od	Numeric vector of size n containing the orthogonal distances to the robust PCA subspace.
cutoff.sd	Cut-off value for the robust score distances.
cutoff.od	Cut-off value for the orthogonal distances.
flag.sd	Numeric vector of size n containing the SD-flags of the observations. The observations whose score distance is larger than cutoff.sd receive an SD-flag equal to zero. The other observations receive an SD-flag equal to 1.
flag.od	Numeric vector of size n containing the OD-flags of the observations. The observations whose orthogonal distance is larger than cutoff.od receive an OD-flag equal to zero. The other observations receive an OD-flag equal to 1.
flag.all	Numeric vector of size n containing the flags of the observations. The observations whose score distance is larger than cutoff.sd or whose orthogonal distance is larger than cutoff.od can be considered as outliers and receive a flag equal to zero. The regular observations receive flag 1.

Author(s)

Tom Reynkens, based on R code from Valentin Todorov for PcaHubert in **rrcov** (released under GPL-3) and Matlab code from Katrien Van Driessen (for the univariate MCD).

References

Hubert, M., Rousseeuw, P. J., and Vanden Branden, K. (2005), "ROBPCA: A New Approach to Robust Principal Component Analysis," *Technometrics*, 47, 64–79.

Engelen, S., Hubert, M. and Vanden Branden, K. (2005), "A Comparison of Three Procedures for Robust PCA in High Dimensions", *Austrian Journal of Statistics*, 34, 117–126.

Hubert, M., Rousseeuw, P. J., and Verdonck, T. (2009), "Robust PCA for Skewed Data and Its Outlier Map," *Computational Statistics & Data Analysis*, 53, 2264–2274.

See Also

PcaHubert, outlyingness, adjOutl

Examples

```
X <- dataGen(m=1, n=100, p=10, eps=0.2, bLength=4)$data[[1]]

resR <- robpca(X, k=2)
diagPlot(resR)
```

---

rospca *RObust Sparse PCA algorithm*

Description

Sparse robust PCA algorithm based on the ROBPCA algorithm of Hubert et al. (2005).

Usage

```
rospca(X, k, kmax = 10, alpha = 0.75, h = NULL, ndir = "all", grid = TRUE,
       lambda = 10^(-6), sparse = "varnum", para, stand = TRUE, skew = FALSE)
```

Arguments

X	An n by p matrix or data matrix with observations in the rows and variables in the columns.
k	Number of principal components that will be used.
kmax	Maximal number of principal components that will be computed, default is 10.
alpha	Robustness parameter, default is 0.75.
h	The number of outliers the algorithm should resist is given by $n - h$. Any value for h between $n/2$ and n may be specified. Default is NULL which uses h=ceiling(alpha*n)+1. Do not specify alpha and h at the same time.
ndir	Number of directions used when computing the outlyingness (or the adjusted outlyingness when skew=TRUE), see outlyingness and adjOutl for more details.
grid	Logical indicating if the grid version of sparse PCA should be used (sPCAgrid with method="sd" from **pcaPP**). Otherwise, the version of Zou et al. (2006) is used (spca from **elasticnet**). Default is TRUE.
lambda	Sparsity parameter of sPCAgrid (when grid=TRUE) or ridge parameter of spca (when grid=FALSE), default is 10^{-6}.
sparse	Parameter for spca (only used when grid=FALSE), see spca for more details.
para	Parameter for spca (only used when grid=FALSE), see spca for more details.
stand	If TRUE, the data are standardised robustly in the beginning and classically before applying sparse PCA. If FALSE, the data are only mean-centred before applying sparse PCA. Default is TRUE.
skew	Logical indicating if the version for skewed data should be applied, default is FALSE.

Details

The ROSPCA algorithm consists of an outlier detection part (step 1), and a sparsification part (steps 2 and 3). We give an overview of these steps here and refer to Hubert et al. (2016) for more details.

Step 1: This is a robustness step similar to ROBPCA. When a standardisation is appropriate, the variables are first robustly standardised by means of the componentwise median and the Q_n. Using the singular value decomposition (SVD) of the resulting data matrix, the p-dimensional data space is reduced to the affine subspace spanned by the n observations. Then, the subset of the h observations with smallest outlyingness is selected (H_0). Thereafter, a reweighting step is applied: given the orthogonal distances to the preliminary PCA subspace determined by the observations in H_0, all observations with orthogonal distances (ODs) smaller than the corresponding cut-off are kept (H_1).

Step 2: First, the data points with indices in H_1 are standardised using the componentwise median and the Q_n and sparse PCA is applied to them. Then, an additional reweighting step is performed which incorporates information about the sparse structure of the data. Variables with zero loadings on all k PCs are discarded and then the orthogonal distances to the estimated sparse PCA subspace are computed. This yields an index set H_2 of observations with orthogonal distance smaller than the cut-off corresponding to these new orthogonal distances. Thereafter, the subset of observations with indices in H_2 is standardised using the componentwise median and the Q_n of the observations in H_1 (the same standardisation as in the first time sparse PCA is applied) and sparse PCA is applied to them which gives sparse loadings. Adding the discarded zero loadings again gives the loadings matrix P_2.

Step 3: In the last step, the eigenvalues are estimated robustly by applying the Q_n^2 estimator on the scores of the observations with indices in H_2. In order to robustly estimate the centre, the score distances are computed and all observations of H_2 with a score distance smaller than the corresponding cut-off are considered, this is the set H_3. Then, the centre is estimated by the mean of these observations. Finally, the estimates of the eigenvalues are recomputed as the sample variance of the (new) scores of the observations with indices in H_3. The eigenvalues are sorted in descending order, so the order of the PCs may change. The columns of the loadings and scores matrices are changed accordingly.

Note that when it is not necessary to standardise the data, they are only centred as in the scheme above, but not scaled.

In contrast to Hubert et al. (2016), we allow for SPCA (Zou et al., 2006) to be used as the sparse PCA method inside ROSPCA (`grid=FALSE`). Moreover, we also include a skew-adjusted version of ROSPCA (`skew=TRUE`) similar to the skew-adjusted version of ROBPCA (Hubert et al., 2009). This adjusted version is not detailed in Hubert et al. (2016).

Value

A list with components:

`loadings`	Loadings matrix containing the sparse robust loadings (eigenvectors), a numeric matrix of size p by k.
`eigenvalues`	Numeric vector of length k containing the robust eigenvalues.
`scores`	Scores matrix (computed as $(X - center) \cdot loadings$), a numeric matrix of size n by k.
`center`	Numeric vector of length k containing the centre of the data.

D	Matrix used to standardise the data before applying sparse PCA (identity matrix if stand=FALSE), a numeric matrix of size p by p.
k	Number of (chosen) principal components.
H0	Logical vector of size n indicating if an observation is in the initial h-subset.
H1	Logical vector of size n indicating if an observation is kept in the non-sparse reweighting step (in robust part).
P1	Loadings matrix before applying sparse reweighting step, a numeric matrix of size p by k.
index	Numeric vector containing the indices of the variables that are used in the sparse reweighting step.
H2	Logical vector of size n indicating if an observation is kept in the sparse reweighting step.
P2	Loadings matrix before estimating eigenvalues, a numeric matrix of size p by k.
H3	Logical vector of size n indicating if an observation is kept in the final SD reweighting step.
alpha	The robustness parameter α used throughout the algorithm.
h	The h-parameter used throughout the algorithm.
sd	Numeric vector of size n containing the robust score distances within the robust PCA subspace.
od	Numeric vector of size n containing the orthogonal distances to the robust PCA subspace.
cutoff.sd	Cut-off value for the robust score distances.
cutoff.od	Cut-off value for the orthogonal distances.
flag.sd	Numeric vector of size n containing the SD-flags of the observations. The observations whose score distance is larger than cutoff.sd receive an SD-flag equal to zero. The other observations receive an SD-flag equal to 1.
flag.od	Numeric vector of size n containing the OD-flags of the observations. The observations whose orthogonal distance is larger than cutoff.od receive an OD-flag equal to zero. The other observations receive an OD-flag equal to 1.
flag.all	Numeric vector of size n containing the flags of the observations. The observations whose score distance is larger than cutoff.sd or whose orthogonal distance is larger than cutoff.od can be considered as outliers and receive a flag equal to zero. The regular observations receive flag 1.

Author(s)

Tom Reynkens, based on R code from Valentin Todorov for PcaHubert in **rrcov** (released under GPL-3) and Matlab code from Katrien Van Driessen (for the univariate MCD).

References

Hubert, M., Reynkens, T., Schmitt, E. and Verdonck, T. (2016). "Sparse PCA for High-Dimensional Data with Outliers," *Technometrics*, 58, 424–434.

Hubert, M., Rousseeuw, P. J., and Vanden Branden, K. (2005), "ROBPCA: A New Approach to Robust Principal Component Analysis," *Technometrics*, 47, 64–79.

Hubert, M., Rousseeuw, P. J., and Verdonck, T. (2009), "Robust PCA for Skewed Data and Its Outlier Map," *Computational Statistics & Data Analysis*, 53, 2264–2274.

Croux, C., Filzmoser, P., and Fritz, H. (2013), "Robust Sparse Principal Component Analysis," *Technometrics*, 55, 202–214.

Zou, H., Hastie, T., and Tibshirani, R. (2006), "Sparse Principal Component Analysis," *Journal of Computational and Graphical Statistics*, 15, 265–286.

See Also

PcaHubert, robpca, outlyingness, adjOutl, sPCAgrid, spca

Examples

```
X <- dataGen(m=1, n=100, p=10, eps=0.2, bLength=4)$data[[1]]

resRS <- rospca(X, k=2, lambda=0.4, stand=TRUE)
diagPlot(resRS)
```

---

selectLambda *Selection of sparsity parameter using IC*

Description

Selection of the sparsity parameter for ROSPCA and SCoTLASS using BIC of Hubert et al. (2016), and for SRPCA using BIC of Croux et al. (2013).

Usage

```
selectLambda(X, k, kmax = 10, method = "ROSPCA", lmin = 0, lmax = 2, lstep = 0.02,
             alpha = 0.75, stand = TRUE, skew = FALSE, multicore = FALSE,
             mc.cores = NULL, P = NULL, ndir = "all")
```

Arguments

X	An n by p matrix or data matrix with observations in the rows and variables in the columns.
k	Number of Principal Components (PCs).
kmax	Maximal number of PCs to be computed, only used when method = "ROSPCA" or method = "ROSPCAg". Default is 10.
method	PCA method to use: ROSPCA ("ROSPCA" or "ROSPCAg"), SCoTLASS ("SCoTLASS" or "SPCAg") or SRPCA ("SRPCA"). Default is "ROSPCA".
lmin	Minimal value of λ to look at, default is 0.
lmax	Maximal value of λ to look at, default is 2.

`lstep`	Difference between two consecutive values of λ, i.e. the step size, default is 0.02.
`alpha`	Robustness parameter for ROSPCA, default is 0.75.
`stand`	Logical indicating if the data should be standardised, default is `TRUE`.
`skew`	Logical indicating if the skewed version of ROSPCA should be applied, default is `FALSE`.
`multicore`	Logical indicating if multiple cores can be used, default is `TRUE`. Note that this is not possible for the Windows platform, so `multicore` is always `FALSE` there.
`mc.cores`	Number of cores to use if `multicore=TRUE`, default is `NULL` which corresponds to the number of cores minus 1.
`P`	True loadings matrix, a numeric matrix of size p by k. The default is `NULL` which means that no true loadings matrix is specified.
`ndir`	Number of directions used when computing the outlyingness (or the adjusted outlyingness when `skew=TRUE`) in `rospca`, see `outlyingness` and `adjOutl` for more details.

Details

We select an optimal value of λ for a certain method on a certain dataset by looking at an equidistant grid of λ values. For each value of λ, we apply the method on the dataset using this sparsity parameter, and compute an Information Criterion (IC). The optimal value of λ is then the one corresponding to the minimal IC. The ICs we consider are the BIC of for Hubert et al. (2016) for ROSPCA and SCoTLASS, and the BIC of Croux et al. (2013) for SRPCA. The BIC of Hubert et al. (2016) is defined as

$$BIC(\lambda) = \ln(1/(h_1 p) \sum_{i=1}^{h_1} OD^2_{(i)}(\lambda)) + df(\lambda) \ln(h_1 p)/(h_1 p),$$

where h_1 is the size of H_1 (the subset of observations that are kept in the non-sparse reweighting step) and $OD_{(i)}(\lambda)$ is the ith smallest orthogonal distance for the model when using λ as the sparsity parameter. The degrees of freedom $df(\lambda)$ are the number of non-zero loadings when λ is used as the sparsity parameter.

Value

A list with components:

`opt.lambda`	Value of λ corresponding to minimal IC.
`min.IC`	Minimal value of IC.
`Lambda`	Numeric vector containing the used values of λ.
`IC`	Numeric cector containing the IC values corresponding to all values of λ in `Lambda`.
`loadings`	Loadings obtained using method with sparsity parameter `opt.lambda`, a numeric matrix of size p by k.

`fit`	Fit obtained using method with sparsity parameter `opt.lambda`. This is a list containing the loadings (`loadings`), the eigenvalues (`eigenvalues`), the standardised data matrix used as input (`Xst`), the scores matrix (`scores`), the orthogonal distances (`od`) and the score distances (`sd`).
`type`	Type of IC used: `BICod` (BIC of Hubert et al. (2016)) or `BIC` (BIC of Croux et al. (2013)).
`measure`	A numeric vector containing the standardised angles between the true and the estimated loadings matrix for each value of λ if a loadings matrix is given. When no loadings matrix is given as input (`P=NULL`), `measure` is equal to `NULL`.

Author(s)

Tom Reynkens

References

Hubert, M., Reynkens, T., Schmitt, E. and Verdonck, T. (2016). "Sparse PCA for High-Dimensional Data with Outliers," *Technometrics*, 58, 424–434.

Croux, C., Filzmoser, P., and Fritz, H. (2013), "Robust Sparse Principal Component Analysis," *Technometrics*, 55, 202–214.

See Also

selectPlot, mclapply, angle

Examples

```
X <- dataGen(m=1, n=100, p=10, eps=0.2, bLength=4)$data[[1]]

sl <- selectLambda(X, k=2, method="ROSPCA", lstep=0.1)
selectPlot(sl)
```

`selectPlot` *Selection plot*

Description

Plot Information Criterion (IC) versus values of the sparsity parameter λ.

Usage

```
selectPlot(sl, indicate = TRUE, main = NULL)
```

Arguments

`sl`	Output from `selectLambda` function.
`indicate`	Logical indicating if the value of λ corresponding to the minimal IC is indicated on the plot, default is `TRUE`.
`main`	Title for the plot, default is `NULL` (no title).

Author(s)

Tom Reynkens

References

Hubert, M., Reynkens, T., Schmitt, E. and Verdonck, T. (2016). "Sparse PCA for High-Dimensional Data with Outliers," *Technometrics*, 58, 424–434.

See Also

selectLambda

Examples

```
X <- dataGen(m=1, n=100, p=10, eps=0.2, bLength=4)$data[[1]]

sl <- selectLambda(X, k=2, method="ROSPCA", lstep=0.1)
selectPlot(sl)
```

---

zeroMeasure *Zero measure*

Description

Compute the average zero measures and total zero measure for a list of matrices.

Usage

```
zeroMeasure(Plist, P, prec = 10^(-5))
```

Arguments

`Plist`	List of estimated loadings matrices or a single estimated loadings matrix. All these matrices should be numeric matrices of size p by k.
`P`	True loadings matrix, a numeric matrix of size p by k.
`prec`	Precision used when determining if an element is non-zero, default is 10^{-5}. We say that all elements with an absolute value smaller than `prec` are "equal to zero".

Details

The *zero measure* is a way to compare how correctly a PCA method estimates the sparse loadings matrix `P`. For each element of an estimated loadings matrix, it is equal to one if the estimated and true value are both zero or both non-zero, and zero otherwise. We then take the average zero measure over all elements of an estimated loadings matrix and over all estimated loadings matrices which we call the *total zero measure*.

Value

A list with components:

`measure`	Numeric matrix of size p by k containing the average zero measure over all `length(Plist)` simulations for each element of `P`.
`index`	Numeric vector containing the indices of all data sets where the estimate was wrong (at least one of the zero measures for the elements of an estimated loadings matrix is equal to 0).
`total`	Total zero measure, i.e. the average zero measure over all elements of an estimated loadings matrix and over all estimated loadings matrices.

Author(s)

Tom Reynkens

References

Hubert, M., Reynkens, T., Schmitt, E. and Verdonck, T. (2016). "Sparse PCA for High-Dimensional Data with Outliers," *Technometrics*, 58, 424–434.

Examples

```
P <- cbind(c(1,1), c(0,1))
Plist <- list(matrix(1,2,2), P)

zeroMeasure(Plist, P)
```

Index