Package ‘ltsspca’

October 9, 2019

Type Package
Title Sparse Principal Component Based on Least Trimmed Squares
Version 0.1.0
Date 2019-09-13
Depends R (>= 3.2.0)
Maintainer Yixin Wang <wangyixin07@outlook.com>
License GPL (>= 2)
Encoding UTF-8
LazyData yes
VignetteBuilder knitr
Suggests robustbase, rrcov, stats, mvtnorm, graphics, knitr, rmarkdown, testthat
Imports Rcpp (>= 1.0.1), pracma
LinkingTo Rcpp, RcppArmadillo
RoxygenNote 6.1.1
NeedsCompilation yes
Author Yixin Wang [aut, cre], Stefan Van Aelst [aut], Holger Cevallos Valdiviezo [ctb] (Original R code for the LTS-PCA algorithm), Tom Reynkens [ctb] (Original R code for angle in the rospca package)
Repository CRAN
Date/Publication 2019-10-09 13:20:02 UTC

R topics documented:

Angle .......................................................... 2
dataSim .......................................................... 3
Angle

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

Usage

Angle(A, B)

Arguments

A numerical matrix of size \( p \) by \( k \)
B numerical matrix of size \( q \) by \( l \)

Value

Standardised last principal angle between A and B.

Author(s)

Tom Reynkens

References


dataSim

Simulate data

description

the function that generates the simulation data set

Usage

dataSim(n = 200, p = 20, bLength = 4, a = c(0.9, 0.5, 0),
SD = c(10, 5, 2), eps = 0, eta = 25, setting = "3", seed = 123,
vc = NULL)

Arguments

n number of observations
p number of variables
bLength the number of correlated variables in the first k blocks
a numeric vector of length k+1 that contains the correlations between the variables in each block (the last block contains uncorrelated variables); by default is (0.9, 0.5, 0)
SD numeric vector of length k+1 that contains the standard deviation of the variables in each block (the last block contains uncorrelated variables); by default is (10, 5, 2)
eps proportion of outliers, default is 0
eta parameter that controls the outlyingness, default is 25
setting type of outliers: setting="1" generates the outliers which are outlying in the first two variables in the second block; setting="2" generates score outliers; setting="3" generates the orthogonal outliers which are easy to detect (the setting used in Hubert, et al (2016)); default is "3"
seed random seed used to simulate the data
vc controls the direction of the score outliers within the PC subspace, default is NULL

Value

a list with components
data generated data matrix
ind row indices of outliers
R Correlation matrix of the data
Sigma Covariance matrix of the data
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) and Hubert et al. (2016).

Usage

Glass

Format

A data frame with columns:

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


References


Examples

```r
## Not run:
data(Glass)
## End(Not run)
```
**ltspca**  
*Principal Component Analysis Based on Least Trimmed Squares (LTS-PCA)*

**Description**

the function that computes LTS-PCA

**Usage**

```r
ltspca(x, q, alpha = 0.5, b.choice = NULL, tol = 1e-06, N1 = 3,
       N2 = 2, N2bis = 10, Npc = 10)
```

**Arguments**

- `x`: the input data matrix
- `q`: the dimension of the PC subspace
- `alpha`: the robust parameter which takes value between 0 to 0.5, default is 0.5
- `b.choice`: initial loading matrix; by default is NULL and the deterministic starting values will be computed by the algorithm
- `tol`: convergence criterion
- `N1`: the number controls the updates for `a` without updating `b` in the concentration step
- `N2`: the number controls outer loop in the concentration step
- `N2bis`: the number controls the outer loop for the selected `b`
- `Npc`: the number controls the inner loop

**Value**

the object of class "ltspca" is returned

- `b`: the unnormalized loading matrix
- `mu`: the center estimate
- `ws`: if the observation is included in the h-subset `ws=1`; otherwise `ws=0`
- `best.cand`: the method which computes the best deterministic starting value in the concentration step

**Author(s)**

Cevallos Valdiviezo

**References**

Examples

```r
## Not run:
ltspcaM <- ltspca(x = x, q = 2, alpha = 0.5)

## End(Not run)
```
The function that computes the reweighted LTS-SPCA.

Usage

\texttt{ltsspcaRw(x, obj, k = NULL, alpha = 0.5, co.sd = 0.25)}

Arguments

\begin{itemize}
  \item \texttt{x} \hspace{1cm} the input data matrix
  \item \texttt{obj} \hspace{1cm} initial LTS-SPCA object given by \texttt{ltsspca} function
  \item \texttt{k} \hspace{1cm} dimension of the PC subspace; by default is NULL then \( k \) takes the value of \( k_{\text{max}} \) in the initial LTS-SPCA
  \item \texttt{alpha} \hspace{1cm} the robust parameter which takes value between 0 to 0.5, default is 0.5
  \item \texttt{co.sd} \hspace{1cm} cutoff value for score outlier weight, default is 0.25
\end{itemize}

Examples

```r
library(mvtnorm)
dataM <- dataSim(n = 200, p = 20, bLength = 4, a = c(0.9, 0.5, 0),
                 SD = c(10, 5, 2), eps = 0, seed = 123)
x <- dataM$data
ltsspcaMI <- ltsspca(x = x, kmax = 5, alpha = 0.5)
ltsspcaMR <- ltsspcaRw(x = x, obj = ltsspcaMI, k = 2, alpha = 0.5)
matplot(ltsspcaMR$loadings,type="b",ylab="Loadings")
```

References

mydiagPlot

Make diagnostic plot using the estimated PC subspace

Description

Make diagnostic plot using the estimated PC subspace

Usage

mydiagPlot(x, obj, k, alpha = 0.5, co.sd = 0.25)

Arguments

x  the input data matrix

obj the returned output from rwltsspcar

k  dimension of the PC subspace

alpha the robust parameter which takes value between 0 to 0.5, default is 0.5

c0.1sd cutoff value for score outlier weight, default is 0.25
Value
the diagnostics of outliers

od the orthogonal distances with respect to the k-dimensional PC subspace
ws.od if the observation is outlying in the orthogonal complement of the PC subspace ws.od=0; otherwise ws.sd=1
co.od the cutoff value for orthogonal distances
sc.wt the score outlier weight, which is compared with 0.25 (by default) to flag score outliers
ws.sd if the observation is outlying with the PC subspace ws.sd=0; otherwise ws.sd=1
co.sd the cutoff value for score outlier weight, default is 0.25
sc.out the returned object when computing the score outlier weights

sPCA_rSVD Sparse Principal Component Analysis via Regularized Singular Value Decomposition (sPCA-rSVD)

Description
the function that computes sPCA_rSVD

Usage
sPCA_rSVD(x, k, method = "hard", center = FALSE, scale = FALSE, l.search = NULL, ls.min = 1)

Arguments
x the input data matrix
k the maximal number of PC’s to search for in the initial stage
method threshold method used in the algorithm; If method = "hard" (default), the hard threshold function is used; if method = "soft", the soft threshold function is used; if method = "scad", the scad threshold function is used
center if center = TRUE the data will be centered by the columnwise means; default is center = FALSE
scale if scale = TRUE the data will be scaled by the columnwise standard deviations; default is scaled = FALSE
l.search a list of length kmax which contains the search grids chosen by the user; default is NULL
ls.min the smallest grid step when searching for the sparsity of each PC; default is 1
Value

an object of class "sPCA_rSVD" is returned

loadings  the sparse loading matrix estimated with sPCA_rSVD
scores    the estimated score matrix
eigenvalues the estimated eigenvalues
spca.it   the list that contains the results of sPCA_rSVD when searching for the individual PCs
ls        the list that contains the final search grid for each PC direction

References


Examples

```r
## Not run:
nonrobM <- sPCA_rSVD(x = x, k = 2, center = T, scale = F)

## End(Not run)
```
Index

*Topic datasets
   Glass, 4
Angle, 2
dataSim, 3
Glass, 4
ltspca, 5
ltssPCA, 6
ltssPCA_Rw, 7
mydiagPlot, 8
sPCA_rSVD, 9