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>

Description Implementation of robust and sparse PCA algorithm of Wang and Van Aelst (2019) <DOI:10.1080/00401706.2019.1671234>.

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` *Standardized 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`	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

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

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

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	numveric 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	numveric 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 contols 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` *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

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.

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

Examples

```
## Not run:
 data(Glass)

## End(Not run)
```

`ltspca` *Principal Component Analysis Based on Least Trimmed Squaers (LTS-PCA)*

Description

the function that computes LTS-PCA

Usage

```
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`	intial 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 in 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

Cevallos Valdiviezo, H., Van Aelst, S. (2019), " Fast computation of robust subspace estimators", *Computational Statistics & Data Analysis*, 134, 171–185.

Examples

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

## End(Not run)
```

`ltsspca` *Sparse Principal Component Analysis Based on Least Trimmed Squaers (LTS-SPCA)*

Description

the function that computes the initial LTS-SPCA

Usage

```
ltsspca(x, kmax, alpha = 0.5, mu.choice = NULL, l.search = NULL,
  ls.min = 1, tol = 1e-06, N1 = 3, N2 = 2, N2bis = 10,
  Npc = 10)
```

Arguments

`x`	the input data matrix
`kmax`	the maximal number of PCs searched by the intial LTS-SPCA
`alpha`	the robust parameter which takes value between 0 to 0.5, default is 0.5
`mu.choice`	the center estimate fixed by the user; by default, the center will be estimated automatically by the algorithm
`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
`tol`	convergence criterion
`N1`	the number controls the updates for a without updating b in the concentration step for LTS-PCA
`N2`	the number controls outer loop in the concentration step for LTS-PCA
`N2bis`	the number controls the outer loop for the selected b for both LTS-PCA and LTS-SPCA
`Npc`	the number controls the inner loop for both LTS-PCA and LTS-SPCA

Value

the object of class "ltsspca" is returned

loadings	the initially estimated loading matrix by LTS-SPCA
mu	the center estimates associated with each PC
spca.it	the list that contains the results of LTS-SPCA when searching for the individual PCs
ls	the list that contains the final search grid for each PC direction

Author(s)

Yixin Wang

References

Wang, Y., Van Aelst, S. (2019), " Sparse Principal Component Based On Least Trimmed Squares", *Technometrics, accepted*.

Examples

```
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")
```

---

ltsspcaRw *Reweighted LTS-SPCA*

Description

the function that computes the reweighted LTS-SPCA

Usage

```
ltsspcaRw(x, obj, k = NULL, alpha = 0.5, co.sd = 0.25)
```

Arguments

x	the input data matrix
obj	initial LTS-SPCA object given by ltsspca function
k	dimension of the PC subspace; by default is NULL then k takes the value of kmax in the initial LTS-SPCA
alpha	the robust parameter which takes value between 0 to 0.5, default is 0.5
co.sd	cutoff value for score outlier weight, default is 0.25

Value

the object of class "ltsspcaRw" is returned

loadings	the sparse loading matrix estimated with reweighted LTS-SPCA
scores	the estimated score matrix
eigenvalues	the estimated eigenvalues
mu	the center estimate
rw.obj	the list that contains the results of sPCA_rSVD on the reduced data
od	the orthonal distances with respect to the initially estimated PC subspace with all the noisy variables removed
co.od	the cutoff value for the orthogonal distances
ws.od	if the observation is outlying in the orthgonal complement of the initially estimated PC subspace ws.od=0; otherwise ws.od=1
sc.wt	the score outlier weight, which is compared with 0.25 (by default) to flag score outliers
co.sd	the cutoff value for score outlier weight, default is 0.25
ws.sd	if the observation is outlying with the PC subspace ws.sd=0; otherwise ws.sd=1
sc.out	the retruned object when computing the score outlier weights

---

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 rwltsspca
k	dimension of the PC subspace
alpha	the robust parameter which takes value between 0 to 0.5, default is 0.5
co.sd	cutoff value for score outlier weight, default is 0.25

Value

the diagnostics of outliers

od	the orthgonal distances with respect to the k-dimensional PC subspace
ws.od	if the observation is outlying in the orthgonal 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 retruned object when computing the score outlier weights

---

sPCA_rSVD *Sparse Principal Component Analysis via Regularized Singular Value Decompsition (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 seach for in the initial stage
method	threshold method used in the algorithm; If method = "hard" (defauls), 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

Shen, H. and Huang, J. (2008), "Sparse principal component anlysis via regularized low rank matrix decomposition", *Journal of Multivariate Analysis*, 99, 1015–1034.

Shen, D., Shen, H., and Marron, J. (2013). "Consistency of sparse PCA in high dimensional low sample size context", *Journal of Multivariate Analysis*, 115, 315–333.

Examples

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

## End(Not run)
```

Index