Package ‘BlockCov’

April 14, 2019

Type Package

Title Estimation of Large Block Covariance Matrices

Version 0.1.1

Author M. Perrot-Dock\{}`es, C. Lévy-Leduc

Maintainer Marie Perrot-Dockès <marie.perrocks@gmail.com>

Description Computation of large covariance matrices having a block structure up to a permutation of their columns and rows
from a small number of samples with respect to the dimension of the matrix.
The method is described in the paper Perrot-Dockès et al. (2019) <arXiv:1806.10093>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

VignetteBuilder knitr

Imports Matrix, stats, Rdpack, BBmisc, dplyr, tibble, magrittr, rlang

Suggests knitr

RdMacros Rdpack

NeedsCompilation no

Repository CRAN

Date/Publication 2019-04-13 22:55:38 UTC

R topics documented:

BlockCov — BlockCov *package*

Description

Estimation of Large Block Covariance Matrices

Details

See the README on CRAN GitHub

---

cv_bl — *Title*

Description

Title

Usage

```
cv_bl(E, v_ord, N)
```

Arguments

E	the observation matrix such that each of its row has a block structure correlation matrix Sigma to estimate up to a permutation of its columns and rows.
v_ord	the absolute value of the upper triangular part matrix Γ (including its diagonal) order in increasing order
N	number of replication in the "cross-validation"

Details

In order to get the treshold one must do rev(v_ord)[cv_bl(E, v_ord, N=N)]

Value

the number of non null values selected for the estimation of the covariance matrix

Examples

```
n <- 30
q <- 100
Sigma <- Simu_Sigma(q = q, diag = FALSE, equal = TRUE)
Matrix::image(Sigma)
E <- matrix(rnorm(n * q), ncol = q) %*% chol(as.matrix(Sigma))
k <- 5
v_up <- est_up(E, k = k)
a_vup <- abs(v_up)
ord_vup <- order(a_vup)
v_ord <- a_vup[ord_vup]
N <- 10
nb_nn0 <- cv_bl(E, v_ord, N=N)
tresh <- rev(v_ord)[nb_nn0]
```

`est_up` *Title*

Description

Title

Usage

```
est_up(E, k = 5)
```

Arguments

`E`	the observation matrix such that each of its row has a block structure correlation matrix Sigma wich has a low rank once its diagonal is removed.
`k`	the rank of the correlation matrix of `E` once its diagonal has been removed

Value

an approximation of the correlation matrix of `E` with its diagonal removed

Examples

```
n <- 30
q <- 100
Sigma <- Simu_Sigma(q = q, diag = FALSE, equal = TRUE)
Matrix::image(Sigma)
E <- matrix(rnorm(n * q), ncol = q) %*% chol(as.matrix(Sigma))
k <- 5
v_up <- est_up(E, k = k)
```

PA *Title*

Description

Title

Usage

```
PA(E, times = 10)
```

Arguments

E	the observation matrix such that each of its row has a block structure correlation matrix Sigma wich has a low rank once its diagonal is removed.
times	number of random sampling

Value

the mean of the eigen values of the `times` sampled matrix

Examples

```
n <- 30
q <- 100
Sigma <- Simu_Sigma(q = q, diag = FALSE, equal = TRUE)
Matrix::image(Sigma)
E <- matrix(rnorm(n * q), ncol = q) %*% chol(as.matrix(Sigma))
random_eigen <- PA(E, times = 10)
```

Sigma_estimation *This function computes an estimator of the covariance matrix and the square root of its inverse and permutes its rows and columns if it is necessary to make the block structure appear.*

Description

This function computes an estimator of the covariance matrix and the square root of its inverse and permutes its rows and columns if it is necessary to make the block structure appear.

Usage

```
Sigma_estimation(E, k = NULL, nb_nn0 = NULL, big = FALSE,
  reorder = FALSE, inv_12 = FALSE, method_k = "Cattell",
  times = 10, method_0 = "Elbow", N = 10)
```

Arguments

`E`	the observation matrix such that each of its row has a block structure correlation matrix Sigma to estimate up to a permutation of its columns and rows.
`k`	numerical or NULL, the rank for the low rank approximation. If NULL the rank is computed using the slope_change function applied on the eigenvalues of the low rank part of Sigma. Default to NULL.
`nb_nn0`	numerical or NULL, corresponds to the number of non null values to keep in the estimation of the covariance matrix. If NULL the number of non null values is computed using the slope_change function to the Frobenius norm of the difference between the empirical correlation matrix and its estimation with nb_nn0 non null values. Default to NULL.
`big`	logical, default to FALSE. If the dataset is too big the empirical correlation is calculated by crossprod(E) * 1 / n to fasten the computation
`reorder`	logical, default to FALSE. Whether or not the columns of E are permuted. If TRUE a hierarchical clustering is first performed and the columns are permuted according to it.
`inv_12`	logical, default to FALSE Whether or not computing the square root of the inverse of the covariance matrix.
`method_k`	character if "Cattell" (the default) then the Cattell criterion (Cattell 1966) is performed on the singular values of the covariance matrix. to estimate the number of rank use in the low rank approximation, while "PA" use the parrallel analysis (Horn 1965) wich can be more accurate if the number of rows of E is not to small but which is much slower.
`times`	numeric the number of resampling done for the "PA" method, ignored if metod_k is different from "PA".
`method_0`	character if "Elbow" (the default) then the Elbow criterion (see Perrot-Dockès et al. (2018) for details) is performed to estimate the number of rank use in the low rank approximation, while "BL" use the approach proposed in Bickel and Levina (2008) based on cross-validation wich can be more accurate if the number of rows of E is not to small but which is much slower.
`N`	numeric the number of fold used for the "BL" method. Ignored if method_0 is different from "BL"

Value

A list with the elements

`Sigma_est`	estimator of the covariance matrix
`k`	rank of the low rank part of the covariance matrix
`nb_nn0`	number of non null values of the upper triangular part of the covariance matrix
`S_inv_12`	square root of the inverse of the estimated covariance matrix
`order`	permutation to apply to the rows and the columns of the covariance to make the block structure appear

References

Bickel PJ, Levina E (2008). “Covariance regularization by thresholding.” *Ann. Statist.*, **36**(6), 2577–2604. doi: 10.1214/08AOS600, https://doi.org/10.1214/08-AOS600.

Cattell RB (1966). “The scree test for the number of factors.” *Multivariate behavioral research*, **1**(2), 245-276.

Horn JL (1965). “A rationale and test for the number of factors in factor analysis.” *Psychometrika*, **30**(2), 179–185. ISSN 1860-0980, doi: 10.1007/BF02289447, https://doi.org/10.1007/BF02289447.

Perrot-Dockès M, Lévy-Leduc C, Rajjou L (2018). “Estimation of large block structured covariance matrices: Application to "multi-omic" approaches to study seed quality.” arXiv:1806.10093.

Examples

```
n <- 30
q <- 100
Sigma <- Simu_Sigma(q = q, diag = FALSE, equal = TRUE)
Matrix::image(Sigma)
E <- matrix(rnorm(n * q), ncol = q) %*% chol(as.matrix(Sigma))
res <- Sigma_estimation(E, inv_12 = TRUE)
Matrix::image(res$Sigma_est)
Matrix::image(res$S_inv_12)
```

Simu_Sigma — *This function generates a block structured symmetric positive definite matrix to test the BlockCov methodology.*

Description

This function generates a block structured symmetric positive definite matrix to test the BlockCov methodology.

Usage

```
Simu_Sigma(q, diag = TRUE, equal = TRUE)
```

Arguments

q	integer corresponding to the size of the covariance matrix.
diag	logical, whether or not the covariance matrix is block-diagonal.
equal	logical, whether or not the values in the blocks are equal.

Value

Sigma a correlation matrix to test the BlockCov methodology.

Examples

```
Sigma <- Simu_Sigma(q = 100, diag = FALSE, equal = TRUE)
Matrix::image(Sigma)
```

---

slope_change	*This function fits to a numerical vector sorted in the non decreasing order two simple linear regressions and returns the index corresponding to the estimated change between the two regression models.*

Description

This function fits to a numerical vector sorted in the non decreasing order two simple linear regressions and returns the index corresponding to the estimated change between the two regression models.

Usage

```
slope_change(Y)
```

Arguments

Y	numerical vector sorted in the non decreasing order.

Value

K the index corresponding to the estimated change between the two linear regression models.

Examples

```
n <- 30
q <- 100
Sigma <- Simu_Sigma(q = q, diag = FALSE, equal = TRUE)
Matrix::image(Sigma)
E <- matrix(rnorm(n * q), ncol = q) %*% chol(as.matrix(Sigma))
corE <- cor(as.matrix(E))
vec_up_emp <- corE[upper.tri(corE)]
G <- matrix(0, ncol = (q - 1), nrow = (q - 1))
G[upper.tri(G, diag = TRUE)] <- vec_up_emp
G[lower.tri(G)] <- t(as.matrix(G))[lower.tri(t(as.matrix(G)))]
res_svd <- svd(G)
vp <- res_svd$d
slope_change(vp)
```

Index