Package 'Support.CCA'

July 2, 2021

Title SUPPORT RECOVERY OF CANONICAL DIRECTIONS IN SPARSE CCA

Version 0.0.0.9000

Description Estimates the support of canonical direction vectors in case of sparse canonical correlation analysis.

License MIT + file LICENSE

Imports mvtnorm, MASS, expm, clime, glasso, glmnet

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

R topics documented:

`ct_support` *Support recovery of the first pair of canonical directions*

Description

This function implements the coordinate thresholding algorithm of *Laha et al. (2020).* to estimate the support of the first pair of canonical directions of X and Y.

Usage

```
ct_support(x, y, sv, t, tau, nl, Sx, Sy, is.standardize)
```

Arguments

x	A matrix with n rows and p columns; corresponds to the first data matrix.
y	A matrix with n rows and q columns; corresponds to the second data matrix.
sv	Optional. A vector giving the number of non-zero elements in the canonical covariates α and β, respectively. See 'details' for more information.
t	Optional. A positive constant corresponding to the threshold level in the co-ordinate thresholding step to estimate the covariance matrix. The soft thresholding of each element of the covariance matrix occurs at $t/\sqrt{n}$ level. If not specified, t is calculated from the data.
tau	Optional. A positive tuning parameter corresponding to the cut-off level in the cleaning step of the co-ordinate thresholding algorithm. If not provided, it is calculated from the data.
nl	Optional. A positive tuning parameter taking value in (0,0.50). Corresponds to the cut-off level in the cleaning step of the co-ordinate thresholding algorithm. If not provided, it is calculated from the data.
Sx	Optional. A $p \times p$ matrix, the variance of X. Must be positive definite. If missing, estimated from the data.
Sy	Optional. A $q \times q$ matrix, the variance of Y. Must be positive definite. If missing, estimated from the data.
is.standardize	Can be either "TRUE" or "FALSE". Indicats hether the columns of x and y are already standardized to have mean zero. The default is FALSE.

Details

sv: sv can be a rough estimate of the sparsity (number of non-zero elements) of α and β, since only the order is important (see laha et al., 2020) here. The returned supports of α and β will not necessarily match the same sparsity levels as provided in sv. In absence of user specified sv, it is estimated from the data using Mai and Zhang (2017)'s SCCA.

tau,nl: The cleaning step of the co-ordinate threshold uses the cut-off level

$$tau\frac{\log(2s)^{0.5+nl}}{s}.$$

Therefore, higher values of tau and nl result in a higher cut-off value, shrinking the estimated support. See Theorem 4 of Laha and Mukherjee (2020) for more details.

Value

A list of two arrays.

- sup.x - An array with length p with binary entries. If the i-th element is 0, it means that i is not in the support of α. Conversely, if the i-th element is 1, it means i is in the support of α.
- sup.y - An array with length p with binary entries. If the i-th element is 0, it means that i is not in the support of β. Conversely, if the i-th element is 1, it means i is in the support of β.

Author(s)

Nilanjana Laha (maintainer), <nlaha@hsph.harvard.edu>, Rajarshi Mukherjee, <ram521@mail.harvard.edu>.

References

Laha, N., Mukherjee, R. (2020) *Support recovery of canonical correlation analysis*. Submitted

Mai, Q., Zhang, X. (2019) *An iterative penalized least squares approach to sparse canonical correlation analysis*, Biometrics, 75, 734-744.

See Also

g_support

Examples

```
library(mvtnorm)
#Simulate  standard normal data matrix: first generate alpha and beta
p <- 500; q <- 200; al <- c(rep(1, 10), rep(0, 490));
be <- c(rep(0,150), rnorm(50,1))

#Normalize alpha and beta
al <- al/sqrt(sum(al^2))
 be <- be/sqrt(sum(be^2))
n <- 300; rho <- 0.5

 #Creating the  covariance matrix
Sigma_mat <- function(p,q,al,be, rho)
{
Sx <- diag(rep(1,p), p, p)
Sy <- diag(rep(1,q), q, q)
Sxy <- tcrossprod(crossprod(rho*Sx, outer(al, be)), Sy)
Syx <- t(Sxy)
rbind(cbind(Sx, Sxy), cbind(Syx, Sy))
}
truesigma <-  Sigma_mat(p,q,al,be, rho)

#Simulating the data
Z <- mvtnorm::rmvnorm(n, sigma = truesigma)
x <- Z[,1:p]
y <- Z[,(p+1):(p+q)]

#Support of alpha
which(c_support(x=x, y=y, sv=c(10, 50))$sup.x==1)
```

`c_support` *Support recovery of the first pair of canonical directions*

Description

This function implements the coordinate thresholding algorithm of *Laha et al. (2020)*. to estimate the support of the first pair of canonical directions of X and Y.

Usage

```
c_support(x, y, sv, c, B, tau, nl, Sx, Sy, is.standardize)
```

Arguments

x	A matrix with n rows and p columns; corresponds to the first data matrix.
y	A matrix with n rows and q columns; corresponds to the second data matrix.
sv	Optional. A vector giving the number of non-zero elements in the canonical covariates α and β, respectively. See 'details' for more information.
c	Optional. A positive constant corresponding to the threshold level in the co-ordinate thresholding step to estimate the covariance matrix. The default choice is one.
B	Optional. The condition number of Σ, the joint covariance matrix of x and y.
tau	Optional. A positive tuning parameter corresponding to the cut-off level in the cleaning step of the co-ordinate thresholding algorithm. If not provided, it is calculated from the data.
nl	Optional. A positive tuning parameter taking value in (0,0.50). Corresponds to the cut-off level in the cleaning step of the co-ordinate thresholding algorithm. If not provided, it is calculated from the data.
Sx	Optional. A $p \times p$ matrix, the variance of X. Must be positive definite. If missing, estimated from the data.
Sy	Optional. A $q \times q$ matrix, the variance of Y. Must be positive definite. If missing, estimated from the data.
is.standardize	Can be either "TRUE" or "FALSE". Indicats hether the variables will be standardized to have mean zero. The default is TRUE.

Details

sv: sv can be a rough estimate of the sparsity (number of non-zero elements) of α and β, since only the order is important (see laha et al., 2020) here. The returned supports of α and β will not necessarily match the same sparsity levels as provided in sv. In absence of user specified sv, it is estimated from the data using Mai and Zhang (2017)'s SCCA.

c1,c2: See Theorem 3 of Laha and Mukherjee (2020) to see how these tuning parameters control the thresholding of the covariance matrix. Larger value of c1 and c2 will result in a higher value of thresholding parameter, which will give a sparser estimator of the covariance matrix. The method is less sensitive to c2 than it is to c1.

tau,nl: The cleaning step of the co-ordinate threshold uses the cut-off level

$$tau\frac{\log(2s)^{0.5+nl}}{s}.$$

Therefore, higher values of tau and nl result in a higher cut-off value, shrinking the estimated support. See Theorem 4 of Laha and Mukherjee (2020) for more details.

Value

A list of two arrays.

- sup.x - An array with length p with binary entries. If the i-th element is 0, it means that i is not in the support of α. Conversely, if the i-th element is 1, it means i is in the support of α.
- sup.y - An array with length p with binary entries. If the i-th element is 0, it means that i is not in the support of β. Conversely, if the i-th element is 1, it means i is in the support of β.

Author(s)

Nilanjana Laha (maintainer), <nlaha@hsph.harvard.edu>, Rajarshi Mukherjee, <ram521@mail.harvard.edu>.

References

Laha, N., Mukherjee, R. (2021) *Support recovery of canonical correlation analysis*. Submitted

Mai, Q., Zhang, X. (2019) *An iterative penalized least squares approach to sparse canonical correlation analysis*, Biometrics, 75, 734-744.

See Also

g_support

Examples

```
library(mvtnorm)
#Simulate  standard normal data matrix: first generate alpha and beta
p <- 500; q <- 200; al <- c(rep(1, 10), rep(0, 490));
be <- c(rep(0,150), rnorm(50,1))

#Normalize alpha and beta
al <- al/sqrt(sum(al^2))
 be <- be/sqrt(sum(be^2))
n <- 300; rho <- 0.5

 #Creating the  covariance matrix
Sigma_mat <- function(p,q,al,be, rho)
{
Sx <- diag(rep(1,p), p, p)
Sy <- diag(rep(1,q), q, q)
Sxy <- tcrossprod(crossprod(rho*Sx, outer(al, be)), Sy)
Syx <- t(Sxy)
rbind(cbind(Sx, Sxy), cbind(Syx, Sy))
}
truesigma <-  Sigma_mat(p,q,al,be, rho)

#Simulating the data
Z <- mvtnorm::rmvnorm(n, sigma = truesigma)
x <- Z[,1:p]
y <- Z[,(p+1):(p+q)]

#Support of alpha
which(c_support(x=x, y=y, sv=c(10, 50), B=1)$sup.x==1)
```

---

g_support *Support recovery of the first pair of canonical directions: unknown sparsity*

Description

Suppose α and β are the first pair of canonical covariates corresponding to random vectors X and Y. This function uses Mai and Zhang (2017)'s SCCA to estimate α and β. The estimation step is followed by a cleaning step, which results in refined estimates of the supports of α and β.

Usage

```
g_support(x, y, Cg, sv, Sx, Sy, is.standardize)
```

Arguments

x	A matrix with n rows and p columns; corresponds to the first data matrix.
y	A matrix with n rows and q columns; corresponds to the second data matrix.
Cg	Optional. A positive constant corresponding to the threshold level in the coordinate thresholding step to estimate the covariance matrix. If not provided, it is calculated from the data.
sv	Optional. A vector giving the number of non-zero elements in the canonical covariates α and β, respectively. See 'details' for more information.
Sx	Optional. A $p \times p$ positive definite matrix, the variance of X. Must be positive definite. If missing, estimated from the data.
Sy	Optional. A $q \times q$ positive definite matrix, the variance of Y. Must be positive definite. If missing, estimated from the data.
is.standardize	Can be either "TRUE" or "FALSE". Indicates whether the variables will be standardized to have mean zero. The default is FALSE.

Details

sv: sv can be a rough estimate of the sparsity (number of non-zero elements) of α and β. The returned supports of α and β will not necessarily match the same sparsity levels as provided in sv. The sparsity is only required for setting a good cut-off in the cleaning step (see Laha and Mukherjee, 2020). In absence of user specified sv, it is estimated from the data using Mai and Zhang (2017)'s SCCA.

Cg: The cut-off in the cleaning step is set by

$$C = Cg \log(2s)/s.$$

See Theorem 2 of Laha and Mukherjee (2020) for more details. Therefore, a higher value of Cg will lead to a smaller estimated support.

Sx,Sy: The method requires an estimator of the inverse covariance matrices, i.e. the precision matrices of X and Y. Unless provided by the user, estimated using glasso method. #of Cai et al. (2011).

Value

A list of two arrays

- sup.x - An array with length p with binary entries. If the i-th element is 0, it means that i is not in the support of α. Conversely, if the i-th element is 1, it means i is in the support of α.
- sup.y - An array with length p with binary entries. If the i-th element is 0, it means that i is not in the support of β. Conversely, if the i-th element is 1, it means i is in the support of β.

Author(s)

Nilanjana Laha (maintainer), <nlaha@hsph.harvard.edu>, Rajarshi Mukherjee, <ram521@mail.harvard.edu>.

References

Laha, N., Mukherjee, R. (2020) *Support recovery of canonical correlation analysis*. Submitted.

Mai, Q., Zhang, X. (2019) *An iterative penalized least squares approach to sparse canonical correlation analysis*, Biometrics, 75, 734-744.

Cai, T., Liu, W., and Luo, X. (2012) *A Constrained l_1 Minimization Approach to Sparse Precision Matrix Estimation*, JASA, 106, 594-607.

See Also

c_support

Examples

```
library(mvtnorm)
#Simulate  standard normal data matrix: first generate alpha and beta
p <- 500; q <- 200; al <- c(rep(1, 10), rep(0, 490));
be <- c(rep(0,150), rnorm(50,1))

#Normalize alpha and beta
al <- al/sqrt(sum(al^2))
 be <- be/sqrt(sum(be^2))
n <- 300; rho <- 0.5

 #Creating the  covariance matrix
Sigma_mat <- function(p,q,al,be, rho)
{
Sx <- diag(rep(1,p), p, p)
Sy <- diag(rep(1,q), q, q)
Sxy <- tcrossprod(crossprod(rho*Sx, outer(al, be)), Sy)
Syx <- t(Sxy)
rbind(cbind(Sx, Sxy), cbind(Syx, Sy))
}
truesigma <-  Sigma_mat(p,q,al,be, rho)

#Simulating the data
Z <- mvtnorm::rmvnorm(n, sigma = truesigma)
x <- Z[,1:p]
y <- Z[,(p+1):(p+q)]

#Support of beta
which(g_support(x,y)$sup.y==1)
```

`m_support` *Support recovery of the first pair of canonical directions using Mai and Zhang (2017)*

Description

Suppose α and β are the first pair of canonical covariates corresponding to random vectors X and Y. This function uses Mai and Zhang (2017)'s SCCA to estimate α and β, and outputs their supports as estimates of the supports of α and β. The code in the authors' website is used to implement the method. This gives an initial estimator of the support, which can be refined by a cleaning step (see Laha and Mukherjee, 2020). The refined estimator of the support is given by the function g_support.

Usage

```
m_support(x, y)
```

Arguments

x	A matrix with n rows and p columns; corresponds to the first data matrix.
y	A matrix with n rows and q columns; corresponds to the second data matrix.

Value

A list of two arrays

- sup.x - An array with length p with binary entries. If the i-th element is 0, it means that i is not in the support of α. Conversely, if the i-th element is 1, it means i is in the support of α.
- sup.y - An array with length p with binary entries. If the i-th element is 0, it means that i is not in the support of β. Conversely, if the i-th element is 1, it means i is in the support of β.

Author(s)

Nilanjana Laha (maintainer), <nlaha@hsph.harvard.edu>, Rajarshi Mukherjee, <ram521@mail.harvard.edu>.

References

Laha, N., Mukherjee, R. (2020) *Support recovery of canonical correlation analysis*. Submitted.

Mai, Q., Zhang, X. (2019) *An iterative penalized least squares approach to sparse canonical correlation analysis*, Biometrics, 75, 734-744.

See Also

g_support

Examples

```
library(mvtnorm)
#Simulate  standard normal data matrix: first generate alpha and beta
p <- 500; q <- 200; al <- c(rep(1, 10), rep(0, 490));
be <- c(rep(0,150), rnorm(50,1))

#Normalize alpha and beta
al <- al/sqrt(sum(al^2))
 be <- be/sqrt(sum(be^2))
n <- 300; rho <- 0.5

 #Creating the  covariance matrix
Sigma_mat <- function(p,q,al,be, rho)
{
Sx <- diag(rep(1,p), p, p)
Sy <- diag(rep(1,q), q, q)
Sxy <- tcrossprod(crossprod(rho*Sx, outer(al, be)), Sy)
Syx <- t(Sxy)
rbind(cbind(Sx, Sxy), cbind(Syx, Sy))
}
truesigma <-  Sigma_mat(p,q,al,be, rho)
```

```
#Simulating the data
Z <- mvtnorm::rmvnorm(n, sigma = truesigma)
x <- Z[,1:p]
y <- Z[,(p+1):(p+q)]

#Support of beta
which(m_support(x,y)$sup.y==1)
```

Index