Package 'mixedCCA'

October 12, 2020

Type Package

Title Sparse Canonical Correlation Analysis for High-Dimensional Mixed Data

Version 1.4.3

Date 2020-10-09

Maintainer Grace Yoon <gyoon6067@gmail.com>

Description Semi-parametric approach for sparse canonical correlation analysis which can handle mixed data types: continuous, binary and truncated continuous. Bridge functions are provided to connect Kendall's tau to latent correlation under the Gaussian copula model. The methods are described in Yoon, Carroll and Gaynanova (2020) <doi:10.1093/biomet/asaa007> and Yoon, Müller and Gaynanova (2020) <arXiv:2006.13875>.

License GPL-3

Encoding UTF-8

Depends R (>= 3.0.1), stats, MASS

Imports Rcpp, pcaPP, Matrix, fMultivar, mnormt, irlba, chebpol

NeedsCompilation yes

RoxygenNote 7.1.1

LinkingTo Rcpp, RcppArmadillo

LazyData true

Author Grace Yoon [aut, cre] (<https://orcid.org/0000-0003-3263-1352>), Irina Gaynanova [aut] (<https://orcid.org/0000-0002-4116-0268>)

Repository CRAN

Date/Publication 2020-10-11 23:40:02 UTC

R topics documented:

`CorrStructure` *Construct a correlation matrix*

Description

Functions to create autocorrelation matrix (p by p) with parameter rho and block correlation matrix (p by p) using group index (of length p) and (possibly) different parameter rho for each group.

Usage

```
autocor(p, rho)

blockcor(blockind, rho)
```

Arguments

`p`	Specified matrix dimension.
`rho`	Correlation value(s), must be between -0.99 and 0.99. Should be a scalar for `autocor`, and either a scalar or a vector of the same length as the maximal `blockind` K for `blockcor`.
`blockind`	Block index 1,. . . , K for a positive integer K specifying which variable belongs to which block, the matrix dimension is equal to `length(blockind)`.

Examples

```
# For p = 8,
# auto correlation matrix
autocor(8, 0.8)
# block correlation matrix: two blocks with the same correlation within each block
blockcor(c(rep(1,3), rep(2,5)), 0.8)
# block correlation matrix: two blocks with different correlation within each block
blockcor(c(rep(1,3), rep(2,5)), c(0.8, 0.3))
```

estimateR *Estimate latent correlation matrix*

Description

Estimation of latent correlation matrix from observed data of (possibly) mixed types (continuous/biary/truncated continuous) based on the latent Gaussian copula model.

Usage

```
estimateR(
  X,
  type = "trunc",
  method = "approx",
  use.nearPD = TRUE,
  nu = 0.01,
  tol = 0.001,
  verbose = FALSE
)

estimateR_mixed(
  X1,
  X2,
  type1 = "trunc",
  type2 = "continuous",
  method = "approx",
  use.nearPD = TRUE,
  nu = 0.01,
  tol = 0.001,
  verbose = FALSE
)
```

Arguments

X	A numeric data matrix (n by p), n is the sample size and p is the number of variables.
type	A type of variables in X, must be one of "continuous", "binary" or "trunc".
method	The calculation method of latent correlation. Either "original" method or "approx". If method = "approx", multilinear approximation method is used, which is much faster than the original method. If method = "original", optimization of the bridge inverse function is used. The default is "approx".
use.nearPD	A logical value indicating whether to use nearPD or not when the resulting correlation estimator is not positive definite (have at least one negative eigenvalue).
nu	Shrinkage parameter for correlation matrix, must be between 0 and 1, the default value is 0.01.
tol	Desired accuracy when calculating the solution of bridge function.

verbose	If verbose = FALSE, printing information whether nearPD is used or not is disabled. The default value is FALSE.
X1	A numeric data matrix (n by p1).
X2	A numeric data matrix (n by p2).
type1	A type of variables in X1, must be one of "continuous", "binary" or "trunc".
type2	A type of variables in X2, must be one of "continuous", "binary" or "trunc".

Value

estimateR returns

- type: Type of the data matrix X
- R: Estimated p by p latent correlation matrix of X

estimateR_mixed returns

- type1: Type of the data matrix X1
- type2: Type of the data matrix X2
- R: Estimated latent correlation matrix of whole X = (X1, X2) (p1+p2 by p1+p2)
- R1: Estimated latent correlation matrix of X1 (p1 by p1)
- R2: Estimated latent correlation matrix of X2 (p2 by p2)
- R12: Estimated latent correlation matrix between X1 and X2 (p1 by p2)

References

Fan J., Liu H., Ning Y. and Zou H. (2017) "High dimensional semiparametric latent graphicalmodel for mixed data" <doi:10.1111/rssb.12168>.

Yoon G., Carroll R.J. and Gaynanova I. (2020) "Sparse semiparametric canonical correlation analysis for data of mixed types" <doi:10.1093/biomet/asaa007>.

Yoon G., Müller C.L., Gaynanova I. (2020) "Fast computation of latent correlations" <arXiv:2006.13875>.

Examples

```
### Data setting
n <- 100; p1 <- 15; p2 <- 10 # sample size and dimensions for two datasets.
maxcancor <- 0.9 # true canonical correlation

### Correlation structure within each data set
set.seed(0)
perm1 <- sample(1:p1, size = p1);
Sigma1 <- autocor(p1, 0.7)[perm1, perm1]
blockind <- sample(1:3, size = p2, replace = TRUE);
Sigma2 <- blockcor(blockind, 0.7)
mu <- rbinom(p1+p2, 1, 0.5)

### true variable indices for each dataset
trueidx1 <- c(rep(1, 3), rep(0, p1-3))
trueidx2 <- c(rep(1, 2), rep(0, p2-2))
```

```
### Data generation
simdata <- GenerateData(n=n, trueidx1 = trueidx1, trueidx2 = trueidx2, maxcancor = maxcancor,
                        Sigma1 = Sigma1, Sigma2 = Sigma2,
                        copula1 = "exp", copula2 = "cube",
                        muZ = mu,
                        type1 = "trunc", type2 = "continuous",
                        c1 = rep(1, p1), c2 =  rep(0, p2)
)
X1 <- simdata$X1
X2 <- simdata$X2

### Check the range of truncation levels of variables
range(colMeans(X1 == 0))
range(colMeans(X2 == 0))

### Estimate latent correlation matrix
# with original method
R1_org <- estimateR(X1, type = "trunc", method = "original")$R
# with faster approximation method
R1_approx <- estimateR(X1, type = "trunc", method = "approx")$R
R12_approx <- estimateR_mixed(X1, X2, type1 = "trunc", type2 = "continuous", method = "approx")$R12
```

`find_w12bic` *Internal mixedCCA function finding w1 and w2 given R1, R2 and R12*

Description

Internal mixedCCA function finding w1 and w2 given R1, R2 and R12

Usage

```
find_w12bic(
  n,
  R1,
  R2,
  R12,
  lamseq1,
  lamseq2,
  w1init,
  w2init,
  BICtype,
  maxiter = 100,
  tol = 0.01,
  trace = FALSE,
  lassoverbose = FALSE
)
```

Arguments

n	Sample size
R1	Correlation matrix of dataset X1 (p1 by p1)
R2	Correlation matrix of dataset X2 (p2 by p2)
R12	Correlation matrix between the dataset X1 and the dataset X2 (p1 by p2)
lamseq1	A sequence of lambda values for the datasets X1. It can be a scalar (a vector of one value). should be the same length with lamseq2.
lamseq2	A sequence of lambda values for the datasets X2. It can be a scalar (a vector of one value). should be the same length with lamseq1.
w1init	An initial vector of length p1 for canonical direction $w1$.
w2init	An initial vector of length p1 for canonical direction $w2$.
BICtype	Either 1 or 2: For more details for two options, see the reference.
maxiter	The maximum number of iterations allowed.
tol	The desired accuracy (convergence tolerance).
trace	If trace = TRUE, progress per each iteration will be printed. The default value is FALSE.
lassoverbose	If lassoverbose = TRUE, all warnings from lassobic optimization regarding convergence will be printed. The default value is lassoverbose = FALSE.

Value

find_w12bic returns a data.frame containing

- w1: estimated canonical direction $w1$.
- w2: estimated canonical direction $w2$.
- w1trace: a matrix, of which column is the estimated canonical direction $w1$ at each iteration. The number of columns is the number of iteration until the convergence.
- w2trace: a matrix, of which column is the estimated canonical direction $w2$ at each iteration. The number of columns is the number of iteration until the convergence.
- lam1.iter: For each iteration, what lambda value is selected for $w1$ is stored.
- lam2.iter: For each iteration, what lambda value is selected for $w2$ is stored.
- obj: objective function value without penalty: $w1^T * R12 * w2$. If lamseq1 and lamseq2 are scalar, then original objective function including penalty part will be used.

References

Yoon G., Carroll R.J. and Gaynanova I. (2020) "Sparse semiparametric canonical correlation analysis for data of mixed types" <doi:10.1093/biomet/asaa007>.

GenerateData *Mixed type simulation data generator for sparse CCA*

Description

GenerateData is used to generate two sets of data of mixed types for sparse CCA under the Gaussian copula model.

Usage

```
GenerateData(
  n,
  trueidx1,
  trueidx2,
  Sigma1,
  Sigma2,
  maxcancor,
  copula1 = "no",
  copula2 = "no",
  type1 = "continuous",
  type2 = "continuous",
  muZ = NULL,
  c1 = NULL,
  c2 = NULL
)
```

Arguments

n	Sample size
trueidx1	True canonical direction of length p1 for X1. It will be automatically normalized such that $w_1^T \Sigma_1 w_1 = 1$.
trueidx2	True canonical direction of length p2 for X2. It will be automatically normalized such that $w_2^T \Sigma_2 w_2 = 1$.
Sigma1	True correlation matrix of latent variable Z1 (p1 by p1).
Sigma2	True correlation matrix of latent variable Z2 (p2 by p2).
maxcancor	True canonical correlation between Z1 and Z2.
copula1	Copula type for the first dataset. U1 = f(Z1), which could be either "exp", "cube".
copula2	Copula type for the second dataset. U2 = f(Z2), which could be either "exp", "cube".
type1	Type of the first dataset X1. Could be "continuous", "trunc" or "binary".
type2	Type of the second dataset X2. Could be "continuous", "trunc" or "binary".
muZ	Mean of latent multivariate normal.

c1	Constant threshold for X1 needed for "trunc" and "binary" data type - the default is NULL.
c2	Constant threshold for X2 needed for "trunc" and "binary" data type - the default is NULL.

Value

GenerateData returns a list containing

- Z1: latent numeric data matrix (n by p1).
- Z2: latent numeric data matrix (n by p2).
- X1: observed numeric data matrix (n by p1).
- X2: observed numeric data matrix (n by p2).
- true_w1: normalized true canonical direction of length p1 for X1.
- true_w2: normalized true canonical direction of length p2 for X2.
- type: a vector containing types of two datasets.
- maxcancor: true canonical correlation between Z1 and Z2.
- c1: constant threshold for X1 for "trunc" and "binary" data type.
- c2: constant threshold for X2 for "trunc" and "binary" data type.
- Sigma: true latent correlation matrix of Z1 and Z2 ((p1+p2) by (p1+p2)).

Examples

```
### Simple example

# Data setting
n <- 100; p1 <- 15; p2 <- 10 # sample size and dimensions for two datasets.
maxcancor <- 0.9 # true canonical correlation

# Correlation structure within each data set
set.seed(0)
perm1 <- sample(1:p1, size = p1);
Sigma1 <- autocor(p1, 0.7)[perm1, perm1]
blockind <- sample(1:3, size = p2, replace = TRUE);
Sigma2 <- blockcor(blockind, 0.7)
mu <- rbinom(p1+p2, 1, 0.5)

# true variable indices for each dataset
trueidx1 <- c(rep(1, 3), rep(0, p1-3))
trueidx2 <- c(rep(1, 2), rep(0, p2-2))

# Data generation
simdata <- GenerateData(n=n, trueidx1 = trueidx1, trueidx2 = trueidx2, maxcancor = maxcancor,
                        Sigma1 = Sigma1, Sigma2 = Sigma2,
                        copula1 = "exp", copula2 = "cube",
                        muZ = mu,
                        type1 = "trunc", type2 = "trunc",
                        c1 = rep(1, p1), c2 =  rep(0, p2)
```

```
)
X1 <- simdata$X1
X2 <- simdata$X2

# Check the range of truncation levels of variables
range(colMeans(X1 == 0))
range(colMeans(X2 == 0))
```

`KendallTau` *Kendall's tau correlation*

Description

Calculate Kendall's tau correlation.

$$\hat{\tau}_{jk} = \frac{2}{n(n-1)} \sum_{1 \le i < i' \le n} sign(X_{ji} - X_{ji'}) sign(X_{ki} - X_{ki'})$$

The function `KendallTau` calculates Kendall's tau correlation between two variables, returning a single correlation value. The function `Kendall_matrix` returns a correlation matrix.

Usage

```
KendallTau(x, y)

Kendall_matrix(X, Y = NULL)
```

Arguments

`x`	A numeric vector.
`y`	A numeric vector.
`X`	A numeric matrix (n by p1).
`Y`	A numeric matrix (n by p2).

Value

`KendallTau(x,y)` returns one Kendall's tau correlation value between two vectors, `x` and `y`.

`Kendall_matrix(X)` returns a p1 by p1 matrix of Kendall's tau correlation coefficients. `Kendall_matrix(X,Y)` returns a p1 by p2 matrix of Kendall's tau correlation coefficients.

Examples

```
n <- 100 # sample size
r <- 0.8 # true correlation

### vector input
# Data generation (X1: truncated continuous, X2: continuous)
```

```
Z <- mvrnorm(n, mu = c(0, 0), Sigma = matrix(c(1, r, r, 1), nrow = 2))
X1 <- Z[,1]
X1[Z[,1] < 1] <- 0
X2 <- Z[,2]

KendallTau(X1, X2)
Kendall_matrix(X1, X2)

### matrix data input
p1 <- 3; p2 <- 4 # dimension of X1 and X2
JSigma <- matrix(r, nrow = p1+p2, ncol = p1+p2); diag(JSigma) <- 1
Z <- mvrnorm(n, mu = rep(0, p1+p2), Sigma = JSigma)
X1 <- Z[,1:p1]
X1[Z[,1:p1] < 0] <- 0
X2 <- Z[,(p1+1):(p1+p2)]

Kendall_matrix(X1, X2)
```

`lambdaseq_generate` *Internal data-driven lambda sequence generating function.*

Description

This internal function generates lambda sequence of length `nlamseq` equally spaced on a logarithmic scale. Since this is for sparse CCA, it returns a list of two vectors. Each vector will be used for each data set $X1$ and $X2$. And $w1$ and $w2$ denote canonical vector for each data set.

Usage

```
lambdaseq_generate(
  nlamseq = 20,
  lam.eps = 0.01,
  Sigma1,
  Sigma2,
  Sigma12,
  w1init = NULL,
  w2init = NULL
)
```

Arguments

`nlamseq`	The length of lambda sequence
`lam.eps`	The smallest value for lambda as a fraction of maximum lambda value
`Sigma1`	Covariance/correlation matrix of $X1$ (p1 by p1)
`Sigma2`	Covariance/correlation matrix of $X2$ (p2 by p2)
`Sigma12`	Covariance/correlation matrix betweem $X1$ and $X2$
`w1init`	Initial value for canonical vector $w1$
`w2init`	Initial value for canonical vector $w2$

Value

lambdaseq_generate returns a list of length 2. Each vector is of the same length nlamseq and will be used for each data set separately.

---

mixedCCA *Sparse CCA for data of mixed types with BIC criterion*

Description

Applies sparse canonical correlation analysis (CCA) for high-dimensional data of mixed types (continuous/biary/truncated continuous). Derived rank-based estimator instead of sample correlation matrix is implemented. There are two types of BIC criteria for variable selection. We found that BIC1 works best for variable selection, whereas BIC2 works best for prediction.

Usage

```
mixedCCA(
  X1,
  X2,
  type1,
  type2,
  lamseq1 = NULL,
  lamseq2 = NULL,
  nlamseq = 20,
  lam.eps = 0.01,
  w1init = NULL,
  w2init = NULL,
  BICtype,
  KendallR = NULL,
  maxiter = 100,
  tol = 0.01,
  trace = FALSE,
  lassoverbose = FALSE
)
```

Arguments

X1	A numeric data matrix (n by p1).
X2	A numeric data matrix (n by p2).
type1	A type of data X1 among "continuous", "binary", "trunc".
type2	A type of data X2 among "continuous", "binary", "trunc".
lamseq1	A tuning parameter sequence for X1. The length should be the same as lamseq2.
lamseq2	A tuning parameter sequence for X2. The length should be the same as lamseq1.
nlamseq	The number of tuning parameter sequence lambda - default is 20.

lam.eps	A ratio of the smallest value for lambda to the maximum value of lambda.
w1init	An initial vector of length p1 for canonical direction $w1$.
w2init	An initial vector of length p2 for canonical direction $w2$.
BICtype	Either 1 or 2: For more details for two options, see the reference.
KendallR	An estimated Kendall τ matrix. The default is NULL, which means that it will be automatically estimated by Kendall's τ estimator unless the user supplies.
maxiter	The maximum number of iterations allowed.
tol	The desired accuracy (convergence tolerance).
trace	If trace = TRUE, progress per each iteration will be printed. The default value is FALSE.
lassoverbose	If lassoverbose = TRUE, all warnings from lassobic optimization regarding convergence will be printed. The default value is lassoverbose = FALSE.

Value

mixedCCA returns a data.frame containing

- KendallR: estimated Kendall's τ matrix estimator.
- lambda_seq: the values of lamseq used for sparse CCA.
- w1: estimated canonical direction $w1$.
- w2: estimated canonical direction $w2$.
- cancor: estimated canonical correlation.
- fitresult: more details regarding the progress at each iteration.

References

Yoon G., Carroll R.J. and Gaynanova I. (2020) "Sparse semiparametric canonical correlation analysis for data of mixed types" <doi:10.1093/biomet/asaa007>.

Examples

```
### Simple example

# Data setting
n <- 100; p1 <- 15; p2 <- 10 # sample size and dimensions for two datasets.
maxcancor <- 0.9 # true canonical correlation

# Correlation structure within each data set
set.seed(0)
perm1 <- sample(1:p1, size = p1);
Sigma1 <- autocor(p1, 0.7)[perm1, perm1]
blockind <- sample(1:3, size = p2, replace = TRUE);
Sigma2 <- blockcor(blockind, 0.7)
mu <- rbinom(p1+p2, 1, 0.5)

# true variable indices for each dataset
trueidx1 <- c(rep(1, 3), rep(0, p1-3))
```

```
trueidx2 <- c(rep(1, 2), rep(0, p2-2))

# Data generation
simdata <- GenerateData(n=n, trueidx1 = trueidx1, trueidx2 = trueidx2, maxcancor = maxcancor,
                        Sigma1 = Sigma1, Sigma2 = Sigma2,
                        copula1 = "exp", copula2 = "cube",
                        muZ = mu,
                        type1 = "trunc", type2 = "trunc",
                        c1 = rep(1, p1), c2 =  rep(0, p2)
)
X1 <- simdata$X1
X2 <- simdata$X2

# Check the range of truncation levels of variables
range(colMeans(X1 == 0))
range(colMeans(X2 == 0))

# Kendall CCA with BIC1
kendallcca1 <- mixedCCA(X1, X2, type1 = "trunc", type2 = "trunc", BICtype = 1)

# Kendall CCA with BIC2. Estimated correlation matrix is plugged in from the above result.
R <- kendallcca1$KendallR
kendallcca2 <- mixedCCA(X1, X2, type1 = "trunc", type2 = "trunc", KendallR = R, BICtype = 2)
```

`myrcc` *Internal RidgeCCA function*

Description

This function is modified from CCA package rcc function. This function is used for simulation. Inputs should be correlation or covariance matrices of each data set and between datasets.

Usage

```
myrcc(R1, R2, R12, lambda1, lambda2, tol = 1e-04)
```

Arguments

`R1`	correlation/covariance/rank-based correlation matrix of dataset `X1`.
`R2`	correlation/covariance/rank-based correlation matrix of dataset `X2`.
`R12`	correlation/covariance/rank-based correlation matrix between dataset `X1` and dataset `X2`.
`lambda1`	tuning parameter (a scalar value) for dataset `X1`.
`lambda2`	tuning parameter (a scalar value) for dataset `X2`.
`tol`	tolerance for eigenvalues. Refer to `standardCCA` function.

Value

myrcc returns a data.frame containing

- cancor: estimated canonical correlation.
- w1: estimated canonical direction $w1$.
- w2: estimated canonical direction $w2$.

---

standardCCA *Internal standard CCA function.*

Description

This function is modified from original CCA function for two reasons: to deal with only positive eigenvalues larger than the tolerance when calculating the inverse of the matrices and to compuate Singular Value Decomposition using irlba algorithm. Inputs should be correlation or covariance matrices of each data set and between datasets. This function returns only the first pair of canonical covariates.

Usage

```
standardCCA(S1, S2, S12, tol = 1e-04)
```

Arguments

S1	correlation/covariance matrix of dataset X1.
S2	correlation/covariance matrix of dataset X2.
S12	correlation/covariance matrix between dataset X1 and dataset X2.
tol	tolerance for eigenvalues. standardCCA function only deals with positive eigenvalues larger than the tolerance.

Value

standardCCA returns a data.frame containing

- cancor: estimated canonical correlation.
- w1: estimated canonical direction $w1$.
- w2: estimated canonical direction $w2$.

Index