Package: GGMridge (via r-universe)

August 21, 2024

Type Package

Title Gaussian Graphical Models Using Ridge Penalty Followed by Thresholding and Reestimation

Version 1.4

Date 2023-11-24

Author Min Jin Ha [aut, cre], Shannon T. Holloway [ctb]

Maintainer Shannon T. Holloway <shannon.t.holloway@gmail.com>

Depends mvtnorm, MASS, stats, graphics

Description Estimation of partial correlation matrix using ridge penalty followed by thresholding and reestimation. Under multivariate Gaussian assumption, the matrix constitutes an Gaussian graphical model (GGM).

License GPL-2

LazyLoad yes

NeedsCompilation no

Encoding UTF-8

RoxygenNote 7.2.1

Collate 'EM.mixture.R' 'scaledMat.R' 'svdFunc.R' 'splitSets.R' 'ne.lambda.cv.R' 'R.separate.ridge.R' 'StructuredEstimate.R' 'ksStat.R' 'getEfronp.R' 'lambda.TargetD.R' 'lambda.cv.R' 'transFisher.R' 'lambda.pcut.cv1.R' 'lambda.pcut.cv.R' 'simulateData.R'

Date/Publication 2023-11-24 22:20:04 UTC

Repository https://sth1402.r-universe.dev

RemoteUrl https://github.com/cran/GGMridge

RemoteRef HEAD

RemoteSha c01a137a5ef20c35f8e5717a170f08e1d9cdb006

Contents

`EM.mixture` *Estimation of the mixture distribution using EM algorithm*

Description

Estimation of the parameters, null proportion, and degrees of freedom of the exact null density in the mixture distribution.

Usage

```
EM.mixture(p, eta0, df, tol)
```

Arguments

`p`	A numeric vector representing partial correlation coefficients.
`eta0`	An initial value for the null proportion; 1-eta0 is the non-null proportion.
`df`	An initial value for the degrees of freedom of the exact null density.
`tol`	The tolerance level for convergence.

Value

A list object containing

`df`	Estimated degrees of freedom of the null density.
`eta0`	Estimated null proportion.
`iter`	The number of iterations required to reach convergence.

Author(s)

Min Jin Ha

References

Schafer, J. and Strimmer, K. (2005). An empirical Bayes approach to inferring large-scale gene association networks. Bioinformatics, 21, 754–764.

`getEfronp` *Estimation of empirical null distribution.*

Description

Estimation of empirical null distribution using Efron's central matching.

Usage

```
getEfronp(
  z,
  bins = 120L,
  maxQ = 9,
  pct = 0,
  pct0 = 0.25,
  cc = 1.2,
  plotIt = FALSE
)
```

Arguments

`z`	A numeric vector of z values following the theoretical normal null distribution.
`bins`	The number of intervals for density estimation of the marginal density of z.
`maxQ`	The maximum degree of the polynomial to be considered for density estimation of the marginal density of z.
`pct`	Low and top (pct*100) f(z).
`pct0`	Low and top (pct0*100) estimate f0(z).
`cc`	The central parts $$(\mu - \sigma cc, \mu + \sigma cc)$$ of the empirical distribution z are used for an estimate of the null proportion (eta).
`plotIt`	TRUE if density plot is to be produced.

Value

A list containing

`correctz`	The corrected z values to follow empirically standard normal distribution.
`correctp`	The corrected p values using the correct z values.
`q`	The chosen degree of polynomial for the estimated marginal density.
`mu0hat`	The location parameter for the normal null distribution.
`sigma0hat`	The scale parameter for the normal null distribution.
`eta`	The estimated null proportion.

Author(s)

Min Jin Ha

References

Efron, B. (2004). Large-scale simultaneous hypothesis testing. Journal of the American Statistical Association, 99, 96–104.

Ha, M. J. and Sun, W. (2014). Partial correlation matrix estimation using ridge penalty followed by thresholding and re-estimation. Biometrics, 70, 762–770.

Examples

```
p <- 100 # number of variables
n <- 50 # sample size

################################
# Simulate data
################################
simulation <- simulateData(G = p, etaA = 0.02, n = n, r = 1)
data <- simulation$data[[1]]
stddata <- scale(x = data, center = TRUE, scale = TRUE)

################################
# estimate ridge parameter
################################
lambda.array <- seq(from = 0.1, to = 20, by = 0.1) * (n - 1.0)
fit <- lambda.cv(x = stddata, lambda = lambda.array, fold = 10L)
lambda <- fit$lambda[which.min(fit$spe)] / (n - 1.0)

################################
# calculate partial correlation
# using ridge inverse
################################
w.upper <- which(upper.tri(diag(p)))
partial <- solve(lambda * diag(p) + cor(data))
partial <- (-scaledMat(x = partial))[w.upper]

################################
# get p-values from empirical
# null distribution
################################
efron.fit <- getEfronp(z = transFisher(x = partial))
```

`ksStat` *The Kolmogorov-Smirnov Statistic for p-Values*

Description

Calculates the Kolmogorov-Smirnov statistic for p-values

Usage

```
ksStat(p)
```

Arguments

p	A numeric vector with p-values.

Value

Kolmogorov-Smirnov statistic

Author(s)

Min Jin Ha

Examples

```
p <- stats::runif(100)
ksStat(p = p)
ks.test(p, y = "punif") # compare with ks.test
```

---

lambda.cv *Choose the Tuning Parameter of the Ridge Inverse*

Description

Choose the tuning parameter of the ridge inverse by minimizing cross validation estimates of the total prediction errors of the p separate ridge regressions.

Usage

```
lambda.cv(x, lambda, fold)
```

Arguments

x	An n by p data matrix.
lambda	A numeric vector of candidate tuning parameters.
fold	fold-cross validation is performed.

Value

A list containing

lambda	The selected tuning parameter, which minimizes the total prediction errors.
spe	The total prediction error for all the candidate lambda values.

Author(s)

Min Jin Ha

References

Ha, M. J. and Sun, W. (2014). Partial correlation matrix estimation using ridge penalty followed by thresholding and re-estimation. Biometrics, 70, 762–770.

Examples

```
p <- 100 # number of variables
n <- 50 # sample size

################################
# Simulate data
################################
simulation <- simulateData(G = p, etaA = 0.02, n = n, r = 1)
data <- simulation$data[[1L]]
stddata <- scale(x = data, center = TRUE, scale = TRUE)

################################
# estimate ridge parameter
################################
lambda.array <- seq(from = 0.1, to = 20, by = 0.1) * (n - 1.0)
fit <- lambda.cv(x = stddata, lambda = lambda.array, fold = 10L)
lambda <- fit$lambda[which.min(fit$spe)] / (n - 1.0)

################################
# calculate partial correlation
# using ridge inverse
################################
partial <- solve(lambda*diag(p) + cor(data))
partial <- -scaledMat(x = partial)
```

`lambda.pcut.cv` *Choose the Tuning Parameter of the Ridge Inverse and Thresholding Level of the Empirical p-Values*

Description

Choose the tuning parameter of the ridge inverse and p-value cutoff by minimizing cross validation estimates of the total prediction errors of the p separate ridge regressions.

Usage

```
lambda.pcut.cv(x, lambda, pcut, fold = 10L)
```

Arguments

x	n by p data matrix.
lambda	A vector of candidate tuning parameters.
pcut	A vector of candidate cutoffs of pvalues.
fold	fold-cross validation is performed.

Value

The total prediction errors for all lambda (row-wise) and pcut (column-wise)

Author(s)

Min Jin Ha

References

Ha, M. J. and Sun, W. (2014). Partial correlation matrix estimation using ridge penalty followed by thresholding and re-estimation. Biometrics, 70, 762–770.

Examples

```
p <- 100 # number of variables
n <- 50 # sample size

################################
# Simulate data
################################
simulation <- simulateData(G = p, etaA = 0.02, n = n, r = 1)
data <- simulation$data[[1L]]
stddata <- scale(x = data, center = TRUE, scale = TRUE)

################################
# Selection of a lambda and a
# p-value cutoff
################################
lambda.array <- seq(from = 0.1, to = 5, length = 10) * (n-1.0)
pcut.array <- seq(from = 0.01, to = 0.05, by = 0.01)
tpe <- lambda.pcut.cv(x = stddata,
                      lambda = lambda.array,
                      pcut = pcut.array,
                      fold = 3L)
w.mintpe <- which(tpe == min(tpe), arr.ind = TRUE)
lambda <- lambda.array[w.mintpe[1L]]
alpha <- pcut.array[w.mintpe[2L]]
```

`lambda.pcut.cv1` *Choose the Tuning Parameter of the Ridge Inverse and Thresholding Level of the Empirical p-Values. Calculate total prediction error for test data after fitting partial correlations from train data for all values of lambda and pcut.*

Description

Choose the Tuning Parameter of the Ridge Inverse and Thresholding Level of the Empirical p-Values.

Calculate total prediction error for test data after fitting partial correlations from train data for all values of lambda and pcut.

Usage

```
lambda.pcut.cv1(train, test, lambda, pcut)
```

Arguments

`train`	An n x p data matrix from which the model is fitted.
`test`	An m x p data matrix from which the model is evaluated.
`lambda`	A vector of candidate tuning parameters.
`pcut`	A vector of candidate cutoffs of pvalues.

Value

Total prediction error for all the candidate lambda and pvalue cutoff values.

Author(s)

Min Jin Ha

References

Ha, M. J. and Sun, W. (2014). Partial correlation matrix estimation using ridge penalty followed by thresholding and re-estimation. Biometrics, 70, 762–770.

Examples

```
p <- 100 # number of variables
n <- 50 # sample size

##############################
# Simulate data
##############################
simulation <- simulateData(G = p, etaA = 0.02, n = n, r = 1)
data <- simulation$data[[1L]]
```

```
##############################
# Split into train/test sets
##############################
testindex <- sample(1L:n, 10L)

train <- data[-testindex,,drop = FALSE]
stdTrain <- scale(x = train, center = TRUE, scale = TRUE)

test <- data[testindex,,drop = FALSE]
stdTest <- scale(x = test, center = TRUE, scale = TRUE)

##############################
# Calculate total prediction
# errors for all candidate
# lambda and p-value cutoffs
##############################
lambda.array <- seq(from = 0.1, to = 5, length = 10) * (n - 1.0)
pcut.array <- seq(from = 0.01, to = 0.05, by = 0.01)
tpe <- lambda.pcut.cv1(train = stdTrain,
                       test = stdTest,
                       lambda = lambda.array,
                       pcut = pcut.array)
```

`lambda.TargetD` *Shrinkage Estimation of a Covariance Matrix Toward an Identity Matrix*

Description

Estimation of a weighted average of a sample covariance (correlation) matrix and an identity matrix.

Usage

```
lambda.TargetD(x)
```

Arguments

`x`	Centered data for covariance shrinkage and standardized data for correlation shrinkage.

Details

An analytical approach to the estimate ridge parameter.

Value

The estimates of shrinkage intensity.

Author(s)

Min Jin Ha

References

Schafer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. Statistical Applications in Genetics and Molecular Biology, 4, 32.

Ha, M. J. and Sun, W. (2014). Partial correlation matrix estimation using ridge penalty followed by thresholding and re-estimation. Biometrics, 70, 762–770.

Examples

```
##############################
 # Simulate data
 ##############################
 simulation <- simulateData(G = 100, etaA = 0.02, n = 50, r = 10)
 dat <- simulation$data[[1L]]
 stddat <- scale(x = dat, center = TRUE, scale = TRUE)

 shrinkage.lambda <- lambda.TargetD(x = stddat)

 ##############################
 # the ridge parameter
 ##############################
 ridge.lambda <- shrinkage.lambda / (1.0 - shrinkage.lambda)

 ##############################
 # partial correlation matrix
 ##############################
 partial <- solve(cor(dat) + ridge.lambda * diag(ncol(dat)))
 partial <- -scaledMat(x = partial)
```

---

ne.lambda.cv — *Choose the Tuning Parameter of a Ridge Regression Using Cross-Validation*

Description

Choose the tuning parameter of a ridge regression using cross-validation.

Usage

```
ne.lambda.cv(y, x, lambda, fold)
```

Arguments

y	Length n response vector.
x	n x p matrix for covariates with p variables and n sample size.
lambda	A numeric vector for candidate tuning parameters for a ridge regression.
fold	fold-cross validation used to choose the tuning parameter.

Value

A list containing

lambda	The selected tuning parameter, which minimizes the prediction error.
spe	The prediction error for all of the candidate lambda values.

Author(s)

Min Jin Ha

References

Ha, M. J. and Sun, W. (2014). Partial correlation matrix estimation using ridge penalty followed by thresholding and re-estimation. Biometrics, 70, 762–770.

Examples

```
p <- 100 # number of variables
n <- 50 # sample size

##############################
# Simulate data
##############################
simulation <- simulateData(G = p, etaA = 0.02, n = n, r = 1)
data <- simulation$data[[1L]]
stddat <- scale(x = data, center = TRUE, scale = TRUE)

X <- stddat[,-1L,drop = FALSE]
y <- stddat[,1L]

fit.lambda <- ne.lambda.cv(y = y,
                           x = X,
                           lambda = seq(from = 0.01, to = 1,by = 0.1),
                           fold = 10L)

lambda <- fit.lambda$lambda[which.min(fit.lambda$spe)]
```

`R.separate.ridge` *Estimation of Partial Correlation Matrix Using p Separate Ridge Regressions.*

Description

The partial correlation matrix is estimated by p separate ridge regressions with the parameters selected by cross validation.

Usage

```
R.separate.ridge(x, fold, lambda, verbose = FALSE)
```

Arguments

`x`	n x p data matrix; n is the # of samples and p is the # of variables.
`fold`	Ridge parameters are selected by fold-cross validations separately for each regression.
`lambda`	The candidate ridge parameters for all p ridge regressions.
`verbose`	TRUE/FALSE; if TRUE, print the procedure.

Value

A list containing

`R`	The partial correlation matrix.
`lambda.sel`	The selected tuning parameters for p ridge regressions.

Author(s)

Min Jin Ha

References

Ha, M. J. and Sun, W. (2014). Partial correlation matrix estimation using ridge penalty followed by thresholding and re-estimation. Biometrics, 70, 762–770.

Examples

```
p <- 100 # number of variables
n <- 50 # sample size

##############################
# Simulate data
##############################
simulation <- simulateData(G = p, etaA = 0.02, n = n, r = 1)
data <- simulation$data[[1L]]
stddata <- scale(x = data, center = TRUE, scale = FALSE)
```

```
##############################
# estimate ridge parameter
##############################
w.upper <- which(upper.tri(diag(p)))

lambda.array <- seq(from = 0.1, to = 20, by=0.1) * (n-1.0)
partial.sep <-  R.separate.ridge(x = stddata,
                                 lambda = lambda.array,
                                 fold = 5L,
                                 verbose = TRUE)$R[w.upper]
```

`scaledMat` *Scale a square matrix*

Description

Scale a square matrix to have unit diagonal elements.

Usage

```
scaledMat(x)
```

Arguments

`x`	A square matrix with positive diagonal elements

Value

Scaled matrix of x.

Author(s)

Min Jin Ha

Examples

```
##############################
# Simulate data
##############################
simulation <- simulateData(G = 100, etaA = 0.02, n = 50, r = 10)
dat <- simulation$data[[1L]]
correlation <- scaledMat(x = stats::cov(dat))
```

simulateData *Generate Simulation Data from a Random Network.*

Description

Generate a random network where both the network structure and the partial correlation coefficients are random. The data matrices are generated from multivariate normal distribution with the covariance matrix corresponding to the network.

Usage

```
simulateData(G, etaA, n, r, dist = "mvnorm")
```

Arguments

G	The number of variables (vertices).
etaA	The proportion of non-null edges among all the G(G-1)/2 edges.
n	The sample size.
r	The number of replicated G by N data matrices.
dist	A function which indicates the distribution of sample. "mvnorm" is multivariate normal distribution and "mvt" is multivariate t distribution with df=2. The default is set by "mvnorm".

Value

A list containing

data	a list, each element containing an n X G matrix of simulated data.
true.partialcor	The partial correlation matrix which the datasets are generated from.
truecor.scaled	The covariance matrix calculted from the partial correlation matrix.
sig.node	The indices of nonzero upper triangle elements of partial correlation matrix.

Author(s)

Min Jin Ha

References

Schafer, J. and Strimmer, K. (2005). An empirical Bayes approach to inferring large-scale gene association networks. Bioinformatics, 21, 754–764.

Examples

```
simulation <- simulateData(G = 100, etaA = 0.02, n = 50, r = 10)
```

`structuredEstimate` *Estimation of Partial Correlation Matrix Given Zero Structure.*

Description

Estimation of nonzero entries of the partial correlation matrix given zero structure.

Usage

```
structuredEstimate(x, E)
```

Arguments

`x`	n by p data matrix with the number of variables p and sample size n.
`E`	The row and column indices of zero entries of the partial correlation matrix.

Value

A list containing

`R`	The partial correlation matrix.
`K`	The inverse covariance matrix.
`RSS`	The residual sum of squares.

Author(s)

Min Jin Ha

References

Ha, M. J. and Sun, W. (2014). Partial correlation matrix estimation using ridge penalty followed by thresholding and re-estimation. Biometrics, 70, 762–770.

Examples

```
p <- 100 # number of variables
n <- 50 # sample size

################################
# Simulate data
################################
simulation <- simulateData(G = p, etaA = 0.02, n = n, r = 1)
data <- simulation$data[[1L]]
stddata <- scale(x = data, center = TRUE, scale = TRUE)

################################
# estimate ridge parameter
################################
```

```
lambda.array <- seq(from = 0.1, to = 20, by = 0.1) * (n-1.0)
fit <- lambda.cv(x = stddata, lambda = lambda.array, fold = 10L)
lambda <- fit$lambda[which.min(fit$spe)]/(n-1)

################################
# calculate partial correlation
# using ridge inverse
################################
w.upper <- which(upper.tri(diag(p)))

partial <- solve(lambda * diag(p) + cor(data))
partial <- (-scaledMat(x = partial))[w.upper]

################################
# get p-values from empirical
# null distribution
################################
efron.fit <- getEfronp(z = transFisher(x = partial),
                       bins = 50L,
                       maxQ = 13)

################################
# estimate the edge set of
# partial correlation graph with
# FDR control at level 0.01
################################
w.array <- which(upper.tri(diag(p)),arr.ind=TRUE)
th <- 0.01
wsig <- which(p.adjust(efron.fit$correctp, method="BH") < th )
E <- w.array[wsig,]
dim(E)

################################
# structured estimation
################################
fit <- structuredEstimate(x = stddata, E = E)
th.partial <- fit$R
```

`transFisher` *Fisher's Z-Transformation*

Description

Fisher's Z-transformation of (partial) correlation.

Usage

```
transFisher(x)
```

Arguments

x	A vector having entries between -1 and 1.

Value

Fisher's Z-transformed values.

Author(s)

Min Jin Ha

Examples

```
################################
# Simulate data
################################
simulation <- simulateData(G = 100, etaA = 0.02, n = 50, r = 1)
dat <- simulation$data[[1L]]
stddat <- scale(x = dat, center = TRUE, scale = TRUE)

shrinkage.lambda <- lambda.TargetD(x = stddat)

################################
# the ridge parameter
################################
ridge.lambda <- shrinkage.lambda / (1.0 - shrinkage.lambda)

################################
# partial correlation matrix
################################
partial <- solve(cor(dat) + ridge.lambda * diag(ncol(dat)))
partial <- -scaledMat(x = partial)

################################
# Fisher's Z transformation of
# upper diagonal of the partial
# correlation matrix
################################
w.upper <- which(upper.tri(diag(nrow(dat))))
psi <- transFisher(x = partial[w.upper])
```

Index