Package 'TOSI'

January 24, 2023

Type Package

Title Two-Directional Simultaneous Inference for High-Dimensional Models

Version 0.2.0

Date 2023-01-23

Author Wei Liu [aut, cre],
Huazhen Lin [aut]

Maintainer Wei Liu <weiliu@smail.swufe.edu.cn>

Description A general framework of two directional simultaneous inference is provided for high-dimensional as well as the fixed dimensional models with manifest variable or latent variable structure, such as high-dimensional mean models, high-dimensional sparse regression models, and high-dimensional latent factors models. It is making the simultaneous inference on a set of parameters from two directions, one is testing whether the estimated zero parameters indeed are zero and the other is testing whether there exists zero in the parameter set of non-zero. More details can be referred to Wei Liu, et al. (2022) <doi:10.48550/arXiv.2012.11100>.

Depends R (>= 4.0.0)

Imports MASS, hdi, scalreg, glmnet

URL https://github.com/feiyoung/TOSI

BugReports https://github.com/feiyoung/TOSI/issues

License GPL

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2023-01-24 10:30:04 UTC

R topics documented:

`assessBsFun` *Assess the performance of group-sparse loading estimate*

Description

Evaluate the model selection consistency rate (SCR), F-measure and the smallest canonical correlation and the larger values mean better peformance in model selection and parameter estimation.

Usage

```
assessBsFun(hB, B0)
```

Arguments

`hB`	a p-by-q matrix, the estimated loading matrix.
`B0`	a p-by-q matrix, the true loading matrix.

Value

return a vecotor with three compoents whose names are scr,fmea, ccorB.

Note

nothing

Author(s)

Liu Wei

See Also

`ccorFun`.

Examples

```
dat <- gendata_Fac(n = 300, p = 500)
res <- gsspFactorm(dat$X)
assessBsFun(res$sphB, dat$B0)
n <- nrow(dat$X)
res <- gsspFactorm(dat$X, lambda1=0.05*n^(1/4), lambda2=9*n^(1/4))
assessBsFun(res$sphB, dat$B0)
```

`bic.spfac` *Modified BIC criteria for selecting penalty parameters*

Description

Evalute the BIC values on a set of grids of penalty parameters.

Usage

```
bic.spfac(X, c1.max= 10, nlamb1=10, C10=4, c2.max=10, nlamb2=10, C20=4)
```

Arguments

`X`	a n-by-p matrix, the observed data
`c1.max`	a positve scalar, the maximum of the grids of c1.
`nlamb1`	a positive integer, the length of grids of penalty parameter lambda1.
`C10`	a positve scalar, the penalty factor C1 of modified BIC.
`c2.max`	a positve scalar, the maximum of the grids of c2.
`nlamb2`	a positive integer, the length of grids of penalty parameter lambda2.
`C20`	a positve scalar, the penalty factor C2 of modified BIC.

Value

return a list with class named `pena_info` and `BIC`, including following components:

`lambda1.min`	a positive number, the penalty value for lambda1 corresponding to the minimum BIC on grids.
`lambda2.min`	a positive number, the penalty value for lambda2 corresponding to the minimum BIC on grids.
`bic1`	a numeric matrix with three columns named c1, lambda1 and bic1, where each row is corresponding to each grid.
`bic2`	a numeric matrix with three columns named c2, lambda2 and bic2, where each row is corresponding to each grid.

Note

nothing

Author(s)

Liu Wei

References

Wei Liu, Huazhen Lin, Jin Liu (2020). Estimation and inference on high-dimensional sparse factor models.

See Also

gsspFactorm.

Examples

```
datlist1 <- gendata_Fac(n= 100, p = 500)
X <- datlist1$X
spfac <- gsspFactorm(X, q=NULL) # use default values for lambda's.
assessBsFun(spfac$sphB, datlist1$B0)

biclist <- bic.spfac(datlist1$X, c2.max=20,nlamb1 = 10) # # select lambda's values using BIC.
plot(biclist$bic1[,2:3], type='o')
plot(biclist$bic2[,2:3], type='o')
spfac <- gsspFactorm(X, q=NULL,biclist$lambda1.min, biclist$lambda2.min)
assessBsFun(spfac$sphB, datlist1$B0)
```

---

ccorFun *Evaluate the smallest canonical correlation for two set of variables*

Description

Evaluate the smallest canonical correlation for two set of variables, each set of variables is represented by a matrix whose columns are variables.

Usage

```
ccorFun(hH, H)
```

Arguments

hH	a n-by-q matrix, one set of q variables.
H	a n-by-q matrix, the other set of q variables.

Value

return a scalar value, the smallest canonical correlation.

Note

nothing

Author(s)

Liu Wei

See Also

cancor.

Examples

```
dat <- gendata_Fac(n = 300, p = 500)
res <- gsspFactorm(dat$X)
ccorFun(res$hH, dat$H0)
```

`cv.spfac` *Cross validation for selecting penalty parameters*

Description

Evalute the CV values on a set of grids of penalty parameters.

Usage

```
cv.spfac(X, lambda1_set, lambda2_set, nfolds=5)
```

Arguments

`X`	a n-by-p matrix, the observed data
`lambda1_set`	a positve vector, the grid for lambda_1.
`lambda2_set`	a positve vector, the grid for lambda_2.
`nfolds`	a positve integer, the folds of cross validation.

Value

return a list including following components:

`lamcv.min`	a 3-dimensional vector, the penalty value for lambda_1 and lambda_2 corresponding to the minimum CV on grids.
`lamcvMat`	a numeric matrix with three columns named lambda_1, lambda_2 and cv, where each row is corresponding to each grid.
`lambda1_set`	the used grid for lambda_1.
`lambda2_set`	the used grid for lambda_2.

Note

nothing

Author(s)

Liu Wei

References

Wei Liu, Huazhen Lin, (2019). Estimation and inference on high-dimensional sparse factor models.

See Also

gsspFactorm.

Examples

```
datlist1 <- gendata_Fac(n= 100, p = 300, rho=1)
X <- datlist1$X
spfac <- gsspFactorm(X, q=NULL)  # use default values for lambda's.
assessBsFun(spfac$sphB, datlist1$B0)
lambda1_set <- seq(0.2, 2, by=0.3)
lambda2_set <- 1:8
# select lambda's values using CV method.
lamList <- cv.spfac(X, lambda1_set, lambda2_set, nfolds=5)
spfac <- gsspFactorm(X, q=NULL,lamList$lamcv.min[1], lamList$lamcv.min[2])
assessBsFun(spfac$sphB, datlist1$B0)
```

FacRowMaxST *Data splitting-based two-stage maximum testing method for a group of loading vectors in factor models.*

Description

Conduct the simultaneous inference for a set of loading vectors in the NUll hypothesises H01 that assumes the set of loading vectors are all zeroes.

Usage

```
FacRowMaxST(X, G1, q=NULL, Nsplit= 5, sub.frac=0.5,
            alpha=0.05, standardized=FALSE,seed=1)
```

Arguments

X	a n-by-p matrix, the observed data
G1	a index set with values of components between 1 and p, the testing set in H01.
q	a positive integer, the number of factors. It will automatically selected by a criterion if it is NULL.
Nsplit	a positive integer, the number of data spliting, default as 5.
sub.frac	a positive number between 0 and 1, the proportion of the sample used in stage I.

alpha	a positive real, the significance level.
standardized	a logical value, whether use the standardized test statistic.
seed	a non-negative integer, the random seed.

Value

return a vector with names 'CriticalValue', 'TestStatistic', 'reject_status', 'p-value' if Nsplit=1, and 'reject_status' and 'adjusted_p-value' if Nsplit>1.

Note

nothing

Author(s)

Liu Wei

References

Wei Liu, Huazhen Lin, Jin Liu (2020). Estimation and inference on high-dimensional sparse factor models.

See Also

Factorm

Examples

```
### Example
dat <- gendata_Fac(n = 300, p = 500)
res <- Factorm(dat$X)
X <- dat$X
# ex1: H01 is false
G1 <- 1:10; # all are nonzero loading vectors
FacRowMaxST(X, G1=G1, alpha=0.05, sub.frac=0.5)
FacRowMaxST(X, q= 6, G1=G1, alpha=0.05, sub.frac=0.5) # specify the true number of factors
# ex2: H01 is true
G1 <- 481:500 # all are zero loading vectors
FacRowMaxST(X, G1=G1, alpha=0.05, sub.frac=0.5)
FacRowMaxST(X, q= 7, G1=G1, alpha=0.05, sub.frac=0.5) # specify a false number of factors
```

FacRowMinST *Data splitting-based two-stage minimum testing method for a group of loading vectors in factor models.*

Description

Conduct the simultaneous inference for a set of loading vectors inr the NUll hypothesises H02 that assumes there is zero loading vector in the set of loading vectors.

Usage

```
FacRowMinST(X, G2,  q=NULL, Nsplit= 5, sub.frac=0.5,
            alpha=0.05, standardized=FALSE,seed=1)
```

Arguments

X	a n-by-p matrix, the observed data
G2	a positive vector with values between 1 and p, the set of H02.
q	a positive integer, the number of factors. It will automatically selected by a criterion if it is NULL.
Nsplit	a positive integer, the number of data spliting, default as 5.
sub.frac	a positive number between 0 and 1, the proportion of the sample used in stage I.
alpha	a positive real, the significance level.
standardized	a logical value, whether use the standardized test statistic.
seed	a non-negative integer, the random seed.

Value

return a vector with names 'CriticalValue', 'TestStatistic', 'reject_status', 'p-value' if Nsplit=1, and 'reject_status' and 'adjusted_p-value' if Nsplit>1.

Note

nothing

Author(s)

Liu Wei

References

Wei Liu, Huazhen Lin, Jin Liu (2020). Estimation and inference on high-dimensional sparse factor models.

See Also

Factorm

Examples

```
### Example
dat <- gendata_Fac(n = 300, p = 500)
res <- Factorm(dat$X)
X <- dat$X
# ex1: H01 is false
G2 <- 1:200; # all are nonzero loading vectors
FacRowMinST(X, G2=G2, alpha=0.05, sub.frac=0.5)
FacRowMinST(X, q= 6, G2=G2, alpha=0.05, sub.frac=0.5) # specify the true number of factors
# ex2: H01 is true
G2 <- 1:500 # all are zero loading vectors
FacRowMinST(X, G2=G2, alpha=0.05, sub.frac=0.5)
FacRowMinST(X, q= 7, G2=G2, alpha=0.05, sub.frac=0.5) # specify a false number of factors
```

`Factorm` *Factor Analysis Model*

Description

Factor analysis to extract latent linear factor and estimate loadings.

Usage

```
Factorm(X, q=NULL)
```

Arguments

`X`	a n-by-p matrix, the observed data
`q`	an integer between 1 and p or `NULL`, default as `NULL` and automatically choose q by the eigenvalue ratio method.

Value

return a list with class named `fac`, including following components:

`hH`	a n-by-q matrix, the extracted lantent factor matrix.
`hB`	a p-by-q matrix, the estimated loading matrix.
`q`	an integer between 1 and p, the number of factor extracted.
`sigma2vec`	a p-dimensional vector, the estimated variance for each error term in model.
`propvar`	a positive number between 0 and 1, the explained propotion of cummulative variance by the q factors.
`egvalues`	a n-dimensional(n<=p) or p-dimensional(p<n) vector, the eigenvalues of sample covariance matrix.

Note

nothing

Author(s)

Liu Wei

References

Fan, J., Xue, L., and Yao, J. (2017). Sufficient forecasting using factor models. Journal of Econometrics.

See Also

factor.

Examples

```
dat <- gendata_Fac(n = 300, p = 500)
res <- Factorm(dat$X)
ccorFun(res$hH, dat$H0) # the smallest canonical correlation
```

---

gendata_Fac *Generate simulated data*

Description

Generate simulated data from high dimensional sparse factor model.

Usage

```
gendata_Fac(n, p, seed=1, q=6, pzero= floor(p/4),
            sigma2=0.1, gamma=1, heter=FALSE, rho=1)
```

Arguments

n	a positive integer, the sample size.
p	an positive integer, the variable dimension.
seed	a nonnegative integer, the random seed, default as 1.
q	a positive integer, the number of factors.
pzero	a positive integer, the number of zero loading vectors, default as p/4.
sigma2	a positive real number, the homogenous variance of error term.
gamma	a positive number, the common component of heteroscedasticity of error term.
heter	a logical value, indicates whether generate heteroscendastic error term.
rho	a positive number, controlling the magnitude of loading matrix.

Value

return a list including two components:

X	a n-by-p matrix, the observed data matrix.
H0	a n-by-q matrix, the true lantent factor matrix.
B0	a p-by-q matrix, the true loading matrix, the last pzero rows are vectors of zeros.
ind_nz	a integer vector, the index vector for which rows of B0 not zeros.

Note

nothing

Author(s)

Liu Wei

See Also

Factorm.

Examples

```
dat <- gendata_Fac(n=300, p = 500)
str(dat)
```

---

gendata_Mean *Generate simulated data*

Description

Generate simulated data from for high-dimensional mean model.

Usage

```
gendata_Mean(n, p, s0= floor(p/2), seed=1, rho= 1, tau=1)
```

Arguments

n	a positive integer, the sample size.
p	an positive integer, the variable dimension.
s0	a positive integer, the number of nonzero components of mean .
seed	a nonnegative integer, the random seed, default as 1.
rho	a positive number between 0 and 1, controlling the correlation of data.
tau	a positive number, controlling the magnitude of covriance matrix.

Value

return a list including two components:

X	a n-by-p matrix, the observed data matrix.
mu	a p-dimensional vector, the mean vector.
p0	a integer vector, the number of nonzero components of mean.

Note

nothing

Author(s)

Liu Wei

Examples

```
dat <- gendata_Mean(n=100, p = 100, s0=3)
str(dat)
```

---

gendata_Reg *Generate simulated data*

Description

Generate simulated data from high-dimensional sparse regression model.

Usage

```
gendata_Reg(n=100, p = 20, s0=5, rho=1, seed=1)
```

Arguments

n	a positive integer, the sample size, default as 100.
p	an positive integer, the dimension of covriates, default as 20.
s0	a positive integer, the number of nonzero components of regression coefficients, default as 5.
rho	a positive number, controlling the magnitude of coefficients.
seed	a nonnegative integer, the random seed, default as 1.

Value

return a list including two components:

Y	a n-dimensional vector, the observed response vector.
X	a n-by-p matrix, the observed covariates matrix.
beta0	a p-dimensional vector, the Reg. coefficients.
index_nz	a integer vector, the index of nonzero components of Reg. coefficients.

Note

nothing

Author(s)

Liu Wei

Examples

```
dat <- gendata_Reg(n=100, p = 100, s0=3)
str(dat)
```

---

gsspFactorm *High Dimensional Sparse Factor Model*

Description

sparse factor analysis to extract latent linear factor and estimate row-sparse and entry-wise-sparse loading matrix.

Usage

```
gsspFactorm(X, q=NULL, lambda1=nrow(X)^(1/4), lambda2=nrow(X)^(1/4))
```

Arguments

X	a n-by-p matrix, the observed data
q	an integer between 1 and p or NULL, default as NULL and automatically choose q by the eigenvalue ratio method.
lambda1	a non-negative number, the row-sparse penalty parameter, default as n^(1/4).
lambda2	a non-negative number, the entry-sparse penalty parameter, default as n^(1/4).

Value

return a list with class named fac, including following components:

hH	a n-by-q matrix, the extracted lantent factor matrix.
sphB	a p-by-q matrix, the estimated row-sparseloading matrix.
hB	a p-by-q matrix, the estimated loading matrix without penalty.
q	an integer between 1 and p, the number of factor extracted.
propvar	a positive number between 0 and 1, the explained propotion of cummulative variance by the q factors.
egvalues	a n-dimensional(n<=p) or p-dimensional(p<n) vector, the eigenvalues of sample covariance matrix.

Note

nothing

Author(s)

Liu Wei

References

Liu, W., Lin, H., Liu, J., & Zheng, S. (2020). Two-directional simultaneous inference for high-dimensional models. arXiv preprint arXiv:2012.11100.

See Also

factor, Factorm

Examples

```
dat <- gendata_Fac(n = 300, p = 500)
res <- gsspFactorm(dat$X)
ccorFun(res$hH, dat$H0) # the smallest canonical correlation
## comparison of l2 norm
oldpar <- par(mar = c(5, 5, 2, 2), mfrow = c(1, 2))
plot(rowSums(dat$B0^2), type='o', ylab='l2B', main='True')
l2B <- rowSums(res$sphB^2)
plot(l2B, type='o', main='Est.')

Bind <- ifelse(dat$B0==0, 0, 1)
hBind <- ifelse(res$sphB==0, 0, 1)

## Select good penalty parameters
dat <- gendata_Fac(n = 300, p = 200)
res <- gsspFactorm(dat$X, lambda1=0.04*nrow(dat$X)^(1/4) ,lambda2=1*nrow(dat$X)^(1/4))
ccorFun(res$hH, dat$H0) # the smallest canonical correlation

## comparison of l2 norm
```

```
plot(rowSums(dat$B0^2), type='o', ylab='l2B', main='True')
l2B <- rowSums(res$sphB^2)
plot(l2B, type='o', main='Est.')

## comparison of structure of loading matrix
Bind <- ifelse(dat$B0==0, 0, 1)
hBind <- ifelse(res$sphB==0, 0, 1)
par(oldpar)
```

MeanMax *Data splitting-based two-stage maximum mean testing method for the mean vector.*

Description

Conduct the simultaneous inference for a set of mean components in the NUll hypothesises H01 that assumes the set of mean components are all zeroes.

Usage

```
MeanMax(X, test.set, Nsplit = 5,frac.size=0.5, standardized=FALSE,alpha=0.05, seed=1)
```

Arguments

X	a n-by-p matrix, the observed data
test.set	a positive vector with values between 1 and p, the set of H01.
Nsplit	a positive integer, the random split times used, default as 5.
frac.size	a positive real between 0 and 1, the proportion of the sample used in stage I.
standardized	a logical value, whether standerdize variables in stage I.
alpha	a positive real, the significant level.
seed	a non-negative integer, the random seed.

Value

return a vector with names 'CriticalValue', 'TestStatistic', 'reject_status', 'p-value' if Nsplit=1, and 'reject_status' and 'adjusted_p-value' if Nsplit>1.

Note

nothing

Author(s)

Liu Wei

See Also

gendata_Mean

Examples

```
### Example
n <- 100; p <- 100;i <- 1
s0 <- 5 # First five components are nonzeros
rho <- 1; tau <- 1;
dat1 <- gendata_Mean(n, p, s0, seed=i, rho, tau)
# ex1: H01 is false
MeanMax(dat1$X, 1:p)
MeanMax(dat1$X, 1:p, Nsplit=1)
# ex1: H01 is true
MeanMax(dat1$X, p)
MeanMax(dat1$X, p, Nsplit=1)
```

MeanMin *Data splitting-based two-stage minimum mean testing method for the mean vector.*

Description

Conduct the simultaneous inference for a set of mean components in the the Null hypothesises H02 that assumes the set of mean components exist zero.

Usage

```
MeanMin(X, test.set, Nsplit = 5, frac.size=0.5, standardized=FALSE, alpha=0.05, seed=1)
```

Arguments

X	a n-by-p matrix, the observed data
test.set	a positive vector with values between 1 and p, the set of H02.
Nsplit	a positive integer, the random split times used, default as 5.
frac.size	a positive number between 0 and 1, the proportion of the sample used in stage I.
standardized	a logical value, whether standerdize in stage I.
alpha	a positive number, the significant level.
seed	a non-negative integer, the random seed.

Value

return a vector with names 'CriticalValue', 'TestStatistic', 'reject_status', 'p-value' if Nsplit=1, and 'reject_status' and 'adjusted_p-value' if Nsplit>1.

Note

nothing

Author(s)

Liu Wei

See Also

gendata_Mean, MeanMin

Examples

```
### Example
n <- 100; p <- 100; i <- 1
s0 <- 5 # First five components are nonzeros
rho <- 4; tau <- 1;
dat1 <- gendata_Mean(n, p, s0, seed=i, rho, tau)
# ex1: H01 is false
MeanMin(dat1$X, 1:s0)
MeanMin(dat1$X, 1:s0, Nsplit=1)
# ex1: H01 is true
MeanMin(dat1$X, 1:p)
MeanMin(dat1$X, 1:p, Nsplit=1)
```

RegMax *Data splitting-based two-stage maximum testing method for the regression coefficients in linear regression models*

Description

Conduct the simultaneous inference for a set of regression coefficients in the null hypothesises H01 that assume the set of regression coefficients components are all zeroes.

Usage

```
RegMax(X, Y, G1, Nsplit = 5, sub.frac=0.5, alpha=0.05, seed=1, standardized=FALSE)
```

Arguments

X	a n-by-p matrix, the observed covariates matrix.
Y	a n-dimensional vector, the observed response vector.
G1	a positive vector with values between 1 and p, the set of H01.
Nsplit	a positive integer, the random split times used, default as 5.
sub.frac	a positive number between 0 and 1, the proportion of the sample used in the stage I.
alpha	a positive real, the significance level.
seed	a non-negative integer, the random seed.
standardized	a logical value, whether standerdize the covariates matrix in the stage I.

Value

return a vector with names 'CriticalValue', 'TestStatistic', 'reject_status', 'p-value' if Nsplit=1, and 'reject_status' and 'adjusted_p-value' if Nsplit>1.

Note

nothing

Author(s)

Liu Wei

References

Liu, W., Lin, H., Liu, J., & Zheng, S. (2020). Two-directional simultaneous inference for high-dimensional models. arXiv preprint arXiv:2012.11100.

See Also

gendata_Reg

Examples

```
### Example
n <- 50; p <- 20; i <- 1
s0 <- 5 # First five components are nonzeros
rho <- 1;
dat1 <- gendata_Reg(n, p, s0, seed=i, rho)
# ex1: H01 is false
RegMax(dat1$X, dat1$Y, 1:p)
# ex1: H01 is true
RegMax(dat1$X, dat1$Y, p)
```

RegMin *Data splitting-based two-Stage minimum testing method for the regression coefficients in linear regression models.*

Description

Conduct the simultaneous inference for a set of regression coefficients in a null hypothesises H02 that assumes the set of regression coefficients components exist zero.

Usage

```
RegMin(X, Y, G2, Nsplit = 5, sub.frac=0.5, alpha=0.05, seed=1, standardized=FALSE)
```

Arguments

X	a n-by-p matrix, the observed covariates matrix.
Y	a n-dimensional vector, the observed outcome vector.
G2	a positive vector with values between 1 and p, the set of regression coefficients in the null hypothesises H02.
Nsplit	a positive integer, the random split times used, default as 5.
sub.frac	a positive number between 0 and 1, the proportion of the sample used in the stage I.
alpha	a positive real, the significance level.
seed	a non-negative integer, the random seed.
standardized	a logical value, whether standerdize the covariates matrix in the stage I.

Value

return a vector with names 'CriticalValue', 'TestStatistic', 'reject_status', 'p-value' if Nsplit=1, and 'reject_status' and 'adjusted_p-value' if Nsplit>1.

Note

nothing

Author(s)

Liu Wei

References

Liu, W., Lin, H., Liu, J., & Zheng, S. (2020). Two-directional simultaneous inference for high-dimensional models. arXiv preprint arXiv:2012.11100.

See Also

gendata_Reg

Examples

```
### Example
n <- 100; p <- 20;i <- 1
s0 <- 5 # First five components are nonzeros
rho <- 1;
dat1 <- gendata_Reg(n, p, s0, seed=i, rho)
# ex1: H01 is false
RegMin(dat1$X, dat1$Y, 1:s0)
# ex1: H01 is true
RegMin(dat1$X, dat1$Y, p)
```

Index