

Package ‘Grace’

January 20, 2025

Type Package

Title Graph-Constrained Estimation and Hypothesis Tests

Version 0.5.3

Date 2017-4-8

Author Sen Zhao

Maintainer Sen Zhao <sen-zhao@sen-zhao.com>

Description Use the graph-constrained estimation (Grace) procedure (Zhao and Shojai, 2016 <[doi:10.1111/biom.12418](https://doi.org/10.1111/biom.12418)>) to estimate graph-guided linear regression coefficients and use the Grace/GraceI/GraceR tests to perform graph-guided hypothesis tests on the association between the response and the predictors.

License GPL-3

Depends MASS, glmnet, scalreg

LazyData TRUE

URL <http://onlinelibrary.wiley.com/doi/10.1111/biom.12418/abstract>

NeedsCompilation no

Repository CRAN

Date/Publication 2017-04-09 21:14:22 UTC

Contents

grace	2
grace.test	3
graceI.test	6
make.L	8

Index	9
--------------	----------

grace

*Graph-Constrained Estimation***Description**

Calculate coefficient estimates of Grace based on methods described in Li and Li (2008).

Usage

```
grace(Y, X, L, lambda.L, lambda.1=0, lambda.2=0, normalize.L=FALSE, K=10, verbose=FALSE)
```

Arguments

Y	outcome vector.
X	matrix of predictors.
L	penalty weight matrix L.
lambda.L	tuning parameter value for the penalty induced by the L matrix (see details). If a sequence of lambda.L values is supplied, K-fold cross-validation is performed.
lambda.1	tuning parameter value for the lasso penalty (see details). If a sequence of lambda.1 values is supplied, K-fold cross-validation is performed.
lambda.2	tuning parameter value for the ridge penalty (see details). If a sequence of lambda.2 values is supplied, K-fold cross-validation is performed.
normalize.L	whether the penalty weight matrix L should be normalized.
K	number of folds in cross-validation.
verbose	whether computation progress should be printed.

Details

The Grace estimator is defined as

$$(\hat{\alpha}, \hat{\beta}) = \arg \min_{\alpha, \beta} \|Y - \alpha 1 - X\beta\|_2^2 + \text{lambda.L}(\beta^T L \beta) + \text{lambda.1}\|\beta\|_1 + \text{lambda.2}\|\beta\|_2^2$$

In the formulation, L is the penalty weight matrix. Tuning parameters lambda.L, lambda.1 and lambda.2 may be chosen by cross-validation. In practice, X and Y are standardized and centered, respectively, before estimating $\hat{\beta}$. The resulting estimate is then rescaled back into the original scale. Note that the intercept $\hat{\alpha}$ is not penalized.

The Grace estimator could be considered as a generalized elastic net estimator (Zou and Hastie, 2005). It penalizes the regression coefficient towards the space spanned by eigenvectors of L with the smallest eigenvalues. Therefore, if L is informative in the sense that $L\beta$ is small, then the Grace estimator could be less biased than the elastic net.

Value

An R 'list' with elements:

intercept	fitted intercept.
beta	fitted regression coefficients.

Author(s)

Sen Zhao

References

Zou, H., and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67, 301-320.

Li, C., and Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24, 1175-1182.

Examples

```
set.seed(120)
n <- 100
p <- 200

L <- matrix(0, nrow = p, ncol = p)
for(i in 1:10){
  L[((i - 1) * p / 10 + 1), ((i - 1) * p / 10 + 1):(i * (p / 10))] <- -1
}
diag(L) <- 0
ind <- lower.tri(L, diag = FALSE)
L[ind] <- t(L)[ind]
diag(L) <- -rowSums(L)

beta <- c(rep(1, 10), rep(0, p - 10))

Sigma <- solve(L + 0.1 * diag(p))
sigma.error <- sqrt(t(beta) %*% Sigma %*% beta) / 2

X <- mvrnorm(n, mu = rep(0, p), Sigma = Sigma)
Y <- c(X %*% beta + rnorm(n, sd = sigma.error))

grace(Y, X, L, lambda.L = c(0.08, 0.12), lambda.2 = c(0.08, 0.12))
```

grace.test

Graph-Constrained Hypothesis Testing Procedure

Description

Test for the association between Y and each column of X using the Grace test based on Zhao and Shojaie (2016).

Usage

```
grace.test(Y, X, L=NULL, lambda.L=NULL, lambda.2=0, normalize.L=FALSE,
eta=0.05, C=4*sqrt(3), K=10, sigma.error=NULL, verbose=FALSE)
```

Arguments

Y	outcome vector.
X	matrix of predictors.
L	penalty weight matrix L .
lambda.L	tuning parameter value for the penalty induced by the L matrix (see details). If a sequence of lambda.L values is supplied, K -fold cross-validation is performed.
lambda.2	tuning parameter value for the ridge penalty (see details). If a sequence of lambda.2 values is supplied, K -fold cross-validation is performed.
normalize.L	whether the penalty weight matrix L should be normalized.
eta	sparsity parameter η (see details).
C	parameter for the initial estimator (see details). It could also be "cv" or "scaled.lasso", in which case cross-validation or the scaled lasso are applied to estimate the initial estimator.
K	number of folds in cross-validation.
sigma.error	error standard deviation. If NULL, scaled lasso is then applied to estimate it.
verbose	whether computation progress should be printed

Details

This function performs the Grace test (if $\lambda.2 = 0$), the GraceI test (if $\lambda.L = 0$) and the GraceR test as introduced in Zhao and Shojaie (2016). The Grace tests examine the null hypothesis $\beta_j = 0$ conditional on all other covariates, even if the design matrix X has more covariates (columns) than observations (rows). Network information on associations between covariates, which is represented by the L matrix, could be used to improve the power of the test. When L is informative (see Zhao and Shojaie (2016) for details), the Grace test is expected to deliver higher power than the GraceI test and other competing methods which ignores the network information (see e.g. Buhlmann (2013), van de Geer et al. (2014), Zhang and Zhang (2014)). When L is potentially uninformative or inaccurate, the GraceR test could be used. Using data-adaptive choices (e.g. by cross-validation) of $\lambda.L$ and $\lambda.2$, the GraceR test could adaptively choose the amount of outside information to be incorporated, which leads to more robust performance. Regardless of the informativeness of L , type-I error rates of the Grace, GraceI and GraceR tests are asymptotically controlled.

The Grace tests are based on the following Grace estimator:

$$(\hat{\alpha}, \hat{\beta}) = \arg \min_{\alpha, \beta} \|Y - \alpha \mathbf{1} - X\beta\|_2^2 + \lambda.L(\beta^T L \beta) + \lambda.2\|\beta\|_2^2$$

In the formulation, L is the penalty weight matrix. Tuning parameters $\lambda.L$ and $\lambda.2$ may be chosen by cross-validation. In practice, X and Y are standardized and centered, respectively, before estimating $\hat{\beta}$. The resulting estimate is then rescaled back into the original scale. Note that the intercept $\hat{\alpha}$ is not penalized.

To perform the Grace tests, the lasso initial estimator is calculated using lasso tuning parameter $\sigma_\epsilon C \sqrt{\log(p)/n}$. The Grace test requires C to be larger than $2\sqrt{2}$. σ_ϵ is the error standard deviation, which is estimated using the scaled lasso (Sun and Zhang, 2012).

$\eta < 0.5$ is the sparsity parameter of β , which controls the level of bias correction. It is assumed that the number of nonzero elements in β , $s = o([n/\log(p)]^\eta)$. This parameter is usually unknown. Using larger values of η leads to more conservative results.

Value

An R 'list' with elements:

intercept	fitted intercept.
beta	fitted regression coefficients.
pvalue	p-values based on the Grace tests.
group.test	function to perform the group test, with the null hypothesis being that all the regression coefficients in the group equal zero. The argument of this function needs to be an index vector of variables. There is an optimal second argument, which specifies whether the group test should be performed based on the "holm" procedure (default), or based on the "max" test statistic. The output is the p-value of the group test. See examples below.

Author(s)

Sen Zhao

References

- Li, C., and Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24, 1175-1182.
- Sun, T., and Zhang, C.-H. (2012). Scaled sparse linear regression. *Biometrika*, 99, 879-898.
- Buhlmann, P. (2013). Statistical significance in high-dimensional linear models. *Bernoulli*, 19, 1212-1242.
- van de Geer, S., Buhlmann, P., Rotiv, Y., and Dezeure, R. (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42, 1166-1202.
- Zhang, C.-H., and Zhang, S.S. (2014). Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B*, 76, 217-242.
- Zhao, S., and Shojaie, A. (2016). A significance test for graph-constrained estimation. *Biometrics*, 72, 484-493.

Examples

```
set.seed(120)
n <- 100
p <- 200

L <- matrix(0, nrow = p, ncol = p)
for(i in 1:10){
```

```

L[((i - 1) * p / 10 + 1), ((i - 1) * p / 10 + 1):(i * (p / 10))] <- -1
}
diag(L) <- 0
ind <- lower.tri(L, diag = FALSE)
L[ind] <- t(L)[ind]
diag(L) <- -rowSums(L)

beta <- c(rep(1, 10), rep(0, p - 10))

Sigma <- solve(L + 0.1 * diag(p))
sigma.error <- sqrt(t(beta) %*% Sigma %*% beta) / 2

X <- mvrnorm(n, mu = rep(0, p), Sigma = Sigma)
Y <- c(X %*% beta + rnorm(n, sd = sigma.error))

grace.test.result <- grace.test(Y, X, L, lambda.L = c(0.08, 0.12),
                                lambda.2 = c(0.08, 0.12))
mean(grace.test.result$pvalue[beta == 0] < 0.05)

grace.test.result$group.test(1:2)
grace.test.result$group.test(3:50)
grace.test.result$group.test(11:100)
grace.test.result$group.test(1:5, method = "max")

grace.test.result <- grace.test(Y, X, L, lambda.L = 0.08,
                                lambda.2 = 0.08, C = "cv")
mean(grace.test.result$pvalue[beta == 0] < 0.05)

```

graceI.test

Graph-Constrained Hypothesis Testing Procedure

Description

Test for the association between Y and each column of X using the GraceI test based on Zhao and Shojaie (2016).

Usage

```
graceI.test(Y,X,lambda.2,eta=0.05,C=4*sqrt(3),K=10,sigma.error=NULL,verbose=FALSE)
```

Arguments

Y	outcome vector.
X	matrix of predictors.
lambda.2	tuning parameter value for the ridge penalty (see details). If a sequence of lambda.2 values is supplied, K-fold cross-validation is performed.
eta	sparsity parameter η (see details).

C	parameter for the initial estimator (see details). It could also be "cv" or "scaled.lasso", in which case cross-validation or the scaled lasso are applied to estimate the initial estimator.
K	number of folds in cross-validation.
sigma.error	error standard deviation. If NULL, scaled lasso is then applied to estimate it.
verbose	whether computation progress should be printed.

Details

This function performs the GraceI test. See the documentation for *grace.test*.

Value

An R 'list' with elements:

intercept	fitted intercept.
beta	fitted regression coefficients.
pvalue	p-values based on the Grace tests.
group.test	function to perform the group test, with the null hypothesis being that all the regression coefficients in the group equal zero. The argument of this function needs to be an index vector of variables. There is an optional second argument, which specifies whether the group test should be performed based on the "holm" procedure (default), or based on the "max" test statistic. The output is the p-value of the group test. See examples below.

Author(s)

Sen Zhao

References

- Li, C., and Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24, 1175-1182.
- Sun, T., and Zhang, C.-H. (2012). Scaled sparse linear regression. *Biometrika*, 99, 879-898.
- Buhlmann, P. (2013). Statistical significance in high-dimensional linear models. *Bernoulli*, 19, 1212-1242.
- van de Geer, S., Buhlmann, P., Rotiv, Y., and Dezeure, R. (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42, 1166-1202.
- Zhang, C.-H., and Zhang, S.S. (2014). Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B*, 76, 217-242.
- Zhao, S., and Shojaie, A. (2016). A significance test for graph-constrained estimation. *Biometrics*, 72, 484-493.

`make.L`*Graph Laplacian Matrix Builder*

Description

This function builds the graph Laplacian matrix from an adjacency matrix.

Usage

```
make.L(adj, normalize.Laplacian=FALSE)
```

Arguments

`adj` adjacency matrix of the graph.

`normalize.Laplacian`

whether the graph Laplacian matrix should be normalized to make all diagonal entries equal to 1. Grace test with the normalized Laplacian matrix is more powerful in identifying hub-covariates.

Value

A matrix object of the graph Laplacian matrix.

Author(s)

Sen Zhao

Index

grace, [2](#)
grace.test, [3](#)
graceI.test, [6](#)
make.L, [8](#)