Package ‘gcdl’

July 31, 2024

Type Package

Title Uncertainty quantification in high-dimensional linear models incorporating graphical structures with applications to gene set analysis

Version 0.1.0

Author Xiao Zhang [aut, cre],
Xiangyong Tan [aut],
Xu Liu [aut]

Maintainer Xiao Zhang <zhangxiao1994@cuhk.edu.cn>

Description Genes function in networks are typically correlated due to their functional connectivity. We construct confidence intervals and provide p-values for parameters of a high-dimensional linear model incorporating graphical structures where the number of variables p diverges as the number of observations. For combining the graphical information, we propose a graph-constrained desparsified LASSO (GCDL) estimator, which reduces dramatically the influence of high correlation of predictors and enjoys the advantage of faster computation and higher accuracy compared with the desparsified LASSO.

License GPL (>= 2)

Imports mnormt, glmnet, glmgraph

Repository github

Encoding UTF-8

LazyData true

LazyDataCompression xz

URL https://github.com/XiaoZhangryy/gcdl

BugReports https://github.com/XiaoZhangryy/gcdl/issues

RoxygenNote 7.3.1

NeedsCompilation yes

R topics documented:

calculate_inv_gram 2
gcdl 2
Laplacian 4
sd_estimator 5
simu_data 6

Index 7

calculate_inv_gram *Calculate Inverse Gram Matrix*

Description

Utilizes the graph structure to compute the approximate inverse of the Gram matrix $x^\top x/n$. This function performs nodewise regression using either ordinary least squares (OLS) or Lasso, depending on the degrees of freedom of the vertices.

Usage

```
calculate_inv_gram(x, G, k = NULL)
```

Arguments

x	The design matrix, which is an n by p matrix, where n is the number of observations and p is the number of predictors.
G	User-specified graph structure matrix. G[i, j] indicates the presence of an edge between nodes i and j.
k	Integer. When the degrees of freedom of the vertex j are less than k, use ordinary least squares; otherwise, use Lasso. Default is NULL, which means k is set to p.

Value

A list containing:

- inverse_Gram - The approximate inverse of the matrix $x^\top x/n$.
- Z - The residuals of the nodewise regressions.

Examples

```
set.seed(0)
data <- simu_data(200, 20, 9)
x <- data$x
G <- data$G
inv_gram <- calculate_inv_gram(x, G)
print(inv_gram$inverse_Gram)
print(inv_gram$Z)
```

gcdl *Calculate P-values based on Graph-Constrained Desparsified LASSO (GCDL) Method*

Description

Calculates P-values based on the graph-constrained desparsified LASSO (GCDL) method. This method incorporates the graph structure into the desparsified LASSO estimator, providing more accurate variable selection in high-dimensional settings.

Usage

```
gcdl(x, y, G, nfolds = 10, Centering = TRUE)
```

Arguments

x	The design matrix, which is an n by p matrix, where n is the number of observations and p is the number of predictors.
y	The response vector, with n elements corresponding to the observations in the design matrix.
G	User-specified graph structure matrix. G[i, j] indicates the presence of an edge between nodes i and j.
nfolds	The number of cross-validation folds. Default is 10.
Centering	Logical. Indicator of whether the design matrix should be centered to column zero mean. Default is TRUE.

Value

A list containing:

- P_value - Individual p-values for each parameter.
- bhat - The GCDL estimator.
- betahat - Initial estimate.
- sigmahat - The estimation of standard deviation obtained through the RCV method.
- Se_bhat - Individual standard deviation for each parameter.
- Inv_Gram - The approximate inverse of the matrix $x^\top x/n$.

References

Chen, L., Liu, H., Kocher, J. P. A., Li, H., & Chen, J. (2015). glmgraph: an R package for variable selection and predictive modeling of structured genomic data. Bioinformatics, 31(24), 3991-3993.

Examples

```
set.seed(0)
data <- simu_data(200, 20, 9)
x <- data$x
y <- data$y
G <- data$G
res <- gcdl(x, y, G)
print(res)
```

Laplacian *Calculate the Laplacian Matrix*

Description

This function computes the Laplacian matrix for a given graph structure matrix.

Usage

```
Laplacian(G)
```

Arguments

G	A square matrix representing the graph structure, where each entry `G[i, j]` indicates the weight of the edge between node `i` and node `j`. If `G[i, j]` is zero, it indicates that there is no edge between the nodes. Diagonal entries are ignored.

Details

The adjacency matrix `A` is derived from the input graph structure matrix `G` by setting `A[i, j] = 1` if `G[i, j] != 0` and `A[i, j] = 0` otherwise. The degree matrix `D` is a diagonal matrix where each diagonal element `D[i, i]` is the sum of the corresponding row in the adjacency matrix `A`.

Value

A square matrix representing the Laplacian matrix `L`. The Laplacian matrix is calculated as `L = D - A`, where `D` is the degree matrix and `A` is the adjacency matrix.

See Also

simu_data

Examples

```
set.seed(0)
data <- simu_data(200, 20, 9)
G <- data$G
lapmat <- Laplacian(G)
print(lapmat)
```

`sd_estimator` *Estimate Standard Deviation*

Description

Utilizes the RCV method to estimate standard deviation. This method is particularly useful in high-dimensional settings where traditional variance estimation methods may not be effective. For more details on the RCV method, see Fan et al. (2012).

Usage

```
sd_estimator(x, y, L, nfolds = 10)
```

Arguments

`x`	The design matrix, which is a `n` by `p` matrix, where `n` is the number of observations and `p` is the number of predictors.
`y`	The response vector, with `n` elements corresponding to the observations in the design matrix.
`L`	Graph Laplacian matrix, which is used to incorporate the graph structure into the estimation process.
`nfolds`	The number of cross-validation folds. Default is 10.

Value

The estimated standard deviation.

References

Fan, J., Guo, S., & Hao, N. (2012). Variance estimation using refitted cross-validation in ultrahigh dimensional regression. Journal of the Royal Statistical Society Series B: Statistical Methodology, 74(1), 37-65.

Examples

```
set.seed(0)
data <- simu_data(200, 20, 9)
x <- data$x
y <- data$y
G <- data$G
lapmat <- Laplacian(G)
sd <- sd_estimator(x, y, lapmat, 10)
print(sd)
```

`simu_data` *Generate the Simulation Data*

Description

Generates simulation data based on a specified covariance structure. This function creates a design matrix, a response vector, and a graph structure matrix.

Usage

```
simu_data(n, p, s0, Cov = NULL)
```

Arguments

`n`	Integer. The sample size.
`p`	Integer. The dimension of the covariates.
`s0`	Integer. The cardinality of the active set. Must be a multiple of 3.
`Cov`	A `p` by `p` positive definite covariance matrix. Default is `NULL`. If not provided, a default covariance matrix with a specific structure is generated.

Value

A list containing:

- `x` - The design matrix, where each row is an observation vector.
- `y` - The response vector.
- `G` - The graph structure matrix.

Examples

```
set.seed(0)
data <- simu_data(200, 20, 9)
x <- data$x
y <- data$y
G <- data$G
```

Index

calculate_inv_gram, 2

gcdl, 2

Laplacian, 4

sd_estimator, 5
simu_data, *4*, 6