Package ‘SDNCMV’

May 25, 2020

Encoding UTF-8

Type Package

Title Simultaneous Differential Network Analysis and Classification
for Matrix-variate Data

Version 1.0.1

Date 2020-05-22

Author Hao Chen, Yong He, Jiadong Ji

Maintainer Hao Chen<chenhaota@163.com>

Depends R (>= 3.5.0), DensParcorr, foreach, doParallel, SIS, stats,
glmnet

Description
This package achieves simultaneous differential network analysis and classification for Matrix-Variate data by the SDNCMV method, see https://arxiv.org/abs/2005.08457.

LazyData yes

License GPL-3

RoxygenNote 7.1.0

NeedsCompilation no

R topics documented:

`SDNCMV-package` *Functions for matrix-variate data classification and identification of differential network.*

Description

SDNCMV provides tools to implement classification and network comparison for matrix-variate data.

References

Hao Chen, Ying Guo, Yong He, Jiadong Ji, Lei Liu, Shi Yufeng, Yikai Wang, Yu Long, Zhang Xinsheng (2020) Simultaneous Differential Network Analysis and Classification for High-dimensional Matrix-variate Data, with application to Brain Connectivity Alteration Detection and fMRI-guided Medical Diagnoses of Alzheimer's Disease. *arXiv e-prints*, arXiv:2005.08457.

Yikai Wang, Jian Kang, Phebe B. Kemmer and Ying Guo (2016) An efficient and reliable statistical method for estimating functional connectivity in large scale brain networks using partial correlation. *Frontiers in Neuroence*, **10**, 123.

Tony Cai,Weidong Liu, and Xi Luo (2011) A constrained ℓ_1 minimization approach for sparse precision matrix estimation. *Journal of the American Statistical Association* **106**, 594-607.

Diego Franco Saldana and Yang Feng (2018) SIS: An R package for Sure Independence Screening in Ultrahigh Dimensional Statistical Models. *Journal of Statistical Software*, **83**, 1-25.

Jianqing Fan and Rui Song (2010) Sure Independence Screening in Generalized Linear Models with NP-Dimensionality. *The Annals of Statistics*, **38**, 3567-3604.

Examples

```
## Not run:
conf_var <- rbind(Conf_var_ADHD, Conf_var_TDC)
data_net <- ParCor_Trans_mat(X1=ADHD, X2=TDC, confounder=TRUE, conf_var=conf_var, p=116, sis_num=1000)

#Classification
# Select 70% samples as training samples in each group, and the remaining samples as test samples.
tr1 <- round(0.7*74)
tr2 <- round(0.7*109)
Y <- data_net$Y_net[c(1:tr1,75:(tr2+75))]
X <- data_net$X_net[c(1:tr1,75:(tr2+75)),]
New_X <- data_net$X_net[c((tr1+1):74,(tr2+76):183),]
prediction <- Classification(X=X,Y=Y,New_X = New_X)

#Identification of Differential Network
Y <- data_net$Y_net
X <- data_net$X_net
p_edge_ind <- data_net$p_edge_ind
delta <- Diff_Net(X=X, Y=Y, p=116, p_edge_ind=p_edge_ind)

## End(Not run)
```

ADHD — *Processed ADHD group data from Peking University site in ADHD-200 Global Competition Dataset*

Description

There are 74 subjects in the ADHD group. For each subject, the data is in a temporal*spatial data matrix form, with spatial dimension 116 (ROIs) and temporal dimension 232.

Usage

```
data(ADHD)
```

Format

A list.

ADHD a list with 74 matrices, and each matrix is of dimension 232*116.

Source

The data set can be downloaded from the website: http://fcon1000.projects.nitrc.org/indi/adhd200/

Classification — *Train bootstrapped Penalized Logistic Regression (PLR) Models with elastic-net penalty and makes prediction for new data.*

Description

`Classification` Train bootstrapped Penalized Logistic Regression (PLR) Models with elastic-net penalty and makes prediction for new data.

Usage

```
Classification(X, Y, New_X, B = 200, alpha = 0.1)
```

Arguments

`X`	Input matrix for bootstrapped Penalized Logistic Regression (PLR) Models. Each row is an observation vector.
`Y`	An binary response vector correspond to matrix X.
`New_X`	Matrix of new values for X at which predictions are to be made.
`B`	Times of bootstrap. Default is 200.
`alpha`	The elastic-net mixing parameter with $0 \leq \alpha \leq 1$. alpha=1 is the lasso penalty, and alpha=0 is the ridge penalty. Default is 0.1.

Details

Classifier by training data X and Y, and then make predictions for new data New_X

Value

An binary outcome for new data New_X.

Author(s)

Hao Chen, Yong He, Jiadong Ji

Examples

```
## Not run:
# Select 70% samples as training samples in each group, and the remaining samples as test samples.
tr1 <- round(0.7*74)
tr2 <- round(0.7*109)
Y <- data_net$Y_net[c(1:tr1,75:(tr2+75))]
X <- data_net$X_net[c(1:tr1,75:(tr2+75)),]
New_X <- data_net$X_net[c((tr1+1):74,(tr2+76):183),]
prediction <- Classification(X=X,Y=Y,New_X = New_X)

## End(Not run)
```

Conf_var_ADHD *Confounders for ADHD subjects in ADHD-200 Global Competition Dataset from Peking University site*

Description

Dataset of three confounders including gender, age and handedness for subjects in the ADHD group.

Usage

```
data(Conf_var_ADHD)
```

Format

A matrix with 74 rows and 3 columns, where the rows correspond to the ADHD subjects and the columns correspond to the confounders.

Source

The data set can be downloaded from the website: http://fcon1000.projects.nitrc.org/indi/adhd200/

Conf_var_TDC *Confounders for TDC subjects in ADHD-200 Global Competition Dataset from Peking University site*

Description

Dataset of three confounders including gender, age and handedness for subjects in the TDC group.

Usage

```
data(Conf_var_TDC)
```

Format

A matrix with 109 rows and 3 columns, where the rows correspond to the TDC subjects and the columns correspond to the confounders.

Source

The data set can be downloaded from the website: http://fcon1000.projects.nitrc.org/indi/adhd200/

Diff_Net *Identification of Differential Network*

Description

`Diff_Net` Identification of Differential Network by a bootstrapped Penalized Logistic Regression (PLR) Models with elastic-net penalty.

Usage

```
Diff_Net(X, Y, B = 200, p, p_edge_ind = NULL, alpha = 0.1)
```

Arguments

X	Input matrix for bootstrapped Penalized Logistic Regression (PLR) Models. Each row is an observation vector.
Y	An binary response vector correspond to matrix X.
B	Times of bootstrap. Default is 200.
p	Row dimension of original matrix-valued data. For example, in fMRI data the p is the number of ROIs.
p_edge_ind	If SIS is called, this parameter is the index of each edge selected by SIS in the previous p*(p-1)/2 edges. If SIS is not called, ignore this parameter. Defalut is `NULL`.
alpha	The elastic-net mixing parameter with $0 \leq \alpha \leq 1$. alpha=1 is the lasso penalty, and alpha=0 is the ridge penalty. Default is 0.1.

Details

Identification of Differential Network of two groups. In this function, first, training the bootstrapped PLR models with elastic-net penalty B times. Then count the number of occurrences of each coefficient $\beta_{i,j}$ and this number also represents the number of occurrences of each pair of nodes (i, j).

Value

A symmetric delta matrix, of dimensions p*p. Each element represents the number of occurrences of each pair of nodes (i, j).

Author(s)

Hao Chen, Yong He, Jiadong Ji

Examples

```
## Not run:
Y <- data_net$Y_net
X <- data_net$X_net
p_edge_ind <- data_net$p_edge_ind
delta <- Diff_Net(X=X, Y=Y, p=116, p_edge_ind=p_edge_ind)

## End(Not run)
```

`ParCor_Trans_mat` *Calculate partial correlations for each subject and do Fisher's Transformation*

Description

`ParCor_Trans_mat` This function first uses R package **DensParcorr** to solve for the partial correlation matrix for each subject, then do Fisher's transformation for the partial correlations. You can use R package **SIS** to screen covariates (edges) first according to specific needs.

Usage

```
ParCor_Trans_mat(
  X1,
  X2,
  confounder = FALSE,
  conf_var = 0,
  p,
  sis_num = NULL,
  cores = NULL,
  dens.level = 0.5
)
```

Arguments

`X1`	A list with n1 elements corresponding to n1 subjects from the case group and each element in this list is a p*q1 matrix.
`X2`	A list with n2 elements corresponding to n2 subjects from the control group and each element in this list is a p*q2 matrix.
`confounder`	logical. If `TRUE`, add confounder variables to final vector data. Default is `FALSE`.
`conf_var`	Confounder matrix, of dimensions n*k, where n=n1+n2, k is the number of confounder variables. If confounder is `TRUE`, then the input confounder matrix will be added; else ignore this parameter.
`p`	Row number of each matrix in X1 and X2.
`sis_num`	Give the variables number s in the data obtained after screening. Default is `NULL`, which means SIS is not called.
`cores`	Give the initial core number for parallel computing. If not specified, the number of cores is set to the number of cores detected by the **parallel** package, if specified, the number of cores is set to the specified value.
`dens.level`	Specify the density level in calculating partial correlation matrix. Default is 0.5. See DensParcorr function in package **DensParcorr** for details.

Details

In detail, we first use the Constrained L1-minimization Approach (CLIME) (Cai et al., 2011) to calculate the partial correlation matrix for each subject and then perform an one-to-one Fisher's transformation on the partial correlations. We then vectorize the upper triangular elements as an observation of network/edge covariates. Sure Independence Screening (SIS) (Fan et al., 2010) method can be called to screen covariates (edges) first. Confounders adjustment can also be taken into account.

Value

a R list from ParCor_Trans_mat containing the following terms:

`X_net`	Network data generated by this function, where each row represents a subject and each column of the first p*(p-1)/2 columns represents an edge and following colunms are confounders Only when sis_num is `NULL`.
`Y_net`	Group label corresponding to each subject in matrix X_net. Only when sis_num is `NULL`.
`X_net_sis`	Network data generated by this function after conducting SIS, where each row represnets a subject and each column of the first sis_num columns represents an edge and following colunms are confounders. Only when sis_num is not `NULL`.
`Y_net_sis`	Group label corresponding to each subject in matrix X_net_sis. Only when sis_num is not `NULL`.
`p_edge_ind`	The index of each edge selected by SIS in the previous p*(p-1)/2 edges

Author(s)

Hao Chen, Yong He, Jiadong Ji

Examples

```
## Not run:
conf_var <- rbind(Conf_var_ADHD, Conf_var_TDC)
data_net <- ParCor_Trans_mat(X1=ADHD, X2=TDC, confounder=TRUE, conf_var=conf_var, p=116, sis_num=1000)

## End(Not run)
```

TDC *Processed TDC group data from Peking University site in ADHD-200 Global Competition Dataset*

Description

There are 109 subjects in the TDC group. For each subject, the data is in a temporal*spatial data matrix form, with spatial dimension 116 (ROIs) and temporal dimension 232.

Usage

```
data(TDC)
```

Format

A list.

TDC a list with 109 matrices, and each matrix is of dimension 232*116.

Source

The data set can be downloaded from the website: http://fcon1000.projects.nitrc.org/indi/adhd200/

Index