Package 'simule'

October 14, 2022

Version 1.3.0

Date 2018-07-02

Title A Constrained L1 Minimization Approach for Estimating Multiple Sparse Gaussian or Nonparanormal Graphical Models

Author Beilun Wang [aut, cre],
Yanjun Qi [aut],
Zhaoyang Wang [aut]

Maintainer Beilun Wang <bw4mw@virginia.edu>

Depends R (>= 3.0.0), lpSolve, pcaPP, igraph

Suggests parallel

Description This is an R implementation of a constrained l1 minimization approach for estimating multiple Sparse Gaussian or Nonparanormal Graphical Models (SIMULE). The SIMULE algorithm can be used to estimate multiple related precision matrices. For instance, it can identify context-specific gene networks from multi-context gene expression datasets. By performing data-driven network inference from high-dimensional and heterogenous data sets, this tool can help users effectively translate aggregated data into knowledge that take the form of graphs among entities. Please run demo(simuleDemo) to learn the basic functions provided by this package. For further details, please read the original paper: Beilun Wang, Ritambhara Singh, Yanjun Qi (2017) <DOI:10.1007/s10994-017-5635-7>.

License GPL-2

URL https://github.com/QData/SIMULE

BugReports https://github.com/QData/SIMULE

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-07-02 21:10:03 UTC

R topics documented:

`simule-package` *Shared and Individual parts of MULtiple graphs Explicitly*

Description

This is an R implementation of a constrained l1 minimization approach for estimating multiple Sparse Gaussian or Nonparanormal Graphical Models (SIMULE). The SIMULE algorithm can be used to estimate multiple related precision matrices. For instance, it can identify context-specific gene networks from multi-context gene expression datasets. By performing data-driven network inference from high-dimensional and heterogeneous datasets, this tool can help users effectively translate aggregated data into knowledge that take the form of graphs among entities. This package includes two graphical model options: Gaussian Graphical model and nonparanormal graphical model. The first model assumes that each dataset follows the Gaussian Distribution. The second one assumes that each dataset is nonparanormal distributed. This package provides two computational options: the multi-threading implementation and the single-threading implementation. Please run `demo(simuleDemo)` to learn the basic functions provided by this package. For further details, please read the original paper: <http://link.springer.com/article/10.1007/s10994-017-5635-7>.

Details

Package:	simule
Type:	Package
Version:	1.3.0
Date:	2018-07-02
License:	GPL (>= 2)

Identifying context-specific entity networks from aggregated data is an important task, often arising in bioinformatics and neuroimaging. Computationally, this task can be formulated as jointly estimating multiple different, but related, sparse Undirected Graphical Models (UGM) from aggregated samples across several contexts. Previous joint-UGM studies have mostly focused on sparse Gaussian Graphical Models (sGGMs) and can't identify context-specific edge patterns directly. We, therefore, propose a novel approach, SIMULE (detecting Shared and Individual parts of MULtiple graphs Explicitly) to learn multi-UGM via a constrained L1 minimization. SIMULE automatically infers both specific edge patterns that are unique to each context and shared interactions preserved among all the contexts. Through the L1 constrained formulation, this problem is cast as multiple

independent subtasks of linear programming that can be solved efficiently in parallel. In addition to Gaussian data, SIMULE can also handle multivariate nonparanormal data that greatly relaxes the normality assumption that many real-world applications do not follow. We provide a novel theoretical proof showing that SIMULE achieves a consistent result at the rate O(log(Kp)/n_tot). On multiple synthetic datasets and two biomedical datasets, SIMULE shows significant improvement over state-of-the-art multi-sGGM and single-UGM baselines.

Author(s)

Beilun Wang, Zhaoyang Wang (Arthur) Beilun Wang(Maintainer)

References

Beilun Wang, Ritambhara Singh, Yanjun Qi (2017). A constrained L1 minimization approach for estimating multiple Sparse Gaussian or Nonparanormal Graphical Models. <http://link.springer.com/article/10.1007/s10994-017-5635-7>

Examples

```
## Not run:
data(exampleData)
simule(X = exampleData , 0.05, 1, covType = "cov", TRUE)
wsimule(X = exampleData , 0.05, 1, W = matrix(1,100,100), covType = "cov", TRUE)

## End(Not run)
```

---

cancer *Microarray data set for breast cancer*

Description

et al's paper. It concerns one hundred thirty-three patients with stage I–III breast cancer. Patients were treated with chemotherapy prior to surgery. Patient response to the treatment can be classified as either a pathologic complete response (pCR) or residual disease (not-pCR). Hess *et al* developed and tested a reliable multigene predictor for treatment response on this data set, composed by a set of 26 genes having a high predictive value.

Usage

```
data(cancer)
```

Format

a list of two objects: dataframe with 133 observations of 26 features and factors indicating whether each sample (out of 133) is of type "not" or type "pcr"

Details

The dataset splits into 2 parts (pCR and not pCR), on which network inference algorithms should be applied independently or in the multitask framework: only individuals from the same classes should be consider as independent and identically distributed.

References

J.A. Mejia, D. Booser, R.L. Theriault, U. Buzdar, P.J. Dempsey, R. Rouzier, N. Sneige, J.S. Ross, T. Vidaurre, H.L. Gomez, G.N. Hortobagyi, and L. Pustzai (2006). Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with Paclitaxel and Fluorouracil, Doxorubicin, and Cyclophosphamide in breast cancer, *Journal of Clinical Oncology*, vol. 24(26), pp. 4236–4244.

`exampleData` *A simulated toy dataset that includes 2 data matrices (from 2 related tasks).*

Description

A simulated toy dataset that includes 2 data matrices (from 2 related tasks). Each data matrix is about 100 features observed in 200 samples. The two data matrices are about exactly the same set of 100 features. This multi-task dataset is generated from two related random graphs. Please run demo(simule) to learn the basic functions provided by this package. For further details, please read the original paper: <http://link.springer.com/article/10.1007/s10994-017-5635-7>.

Usage

```
data(exampleData)
```

Format

The format is: List of 2 matrices $: num [1:200, 1:100] -0.0982 -0.2417 -1.704 0.4 ... ..- attr(*, "dimnames")=List of 2 .. ..$: NULL .. ..$: NULL $: num [1:200, 1:100] -0.161 0.41 0.17 0. ... ..- attr(*, "dimnames")=List of 2 .. ..$: NULL .. ..$: NULL

`exampleDataGraph` *A simulated toy dataset that includes 3 igraph objects*

Description

(first one being the shared graph and second and third being task specific 1 and 2 graphs) The graphs are generated from two related random graphs and the underlaying high dimensional gaussian distribution generates the exampleData dataset. exampleDataGraph serves as a groundtruth to compare in demo(synthetic).

Usage

```
data(exampleDataGraph)
```

Format

A list of 3 igraph objects

---

nip_37_data *NIPS word count dataset*

Description

This NIPS Conference Papers 1987-2015 Data set is avaiable at UCI Machine Learning Repository. The original dataset is in the form of a 11463 x 5812 matrix of word counts (11463 words and 5812 conference papers) Due to the size of the original dataset, it is preprocessed and reduced to a list of two matrices (2900 x 37 and 2911 x 37) The dataset consists of two tasks (early (up to 2006) and recent (after 2006) NIPS conference papers) with 37 words

Usage

```
data(nip_37_data)
```

Format

a list of two nonnegative integer matrices (1:2900, 1:37) and (1:2911,1:37) Columns are named with year_paperid and rows are names with word name

References

'Poisson Random Fields for Dynamic Feature Models'. Perrone V., Jenkins P. A., Spano D., Teh Y. W. (2016)

---

plot.simule *Plot simule result specified by user input*

Description

This function can plot and return multiple sparse graphs distinguished by edge colors from the result generated by simule

Usage

```
## S3 method for class 'simule'
plot(x, graphlabel = NULL, type = "task",
  neighbouroption = "task", subID = NULL, index = NULL,
  graphlayout = NULL, ...)
```

Arguments

x	output generated from simule/wsimule function (simule/wsimule class)
graphlabel	vertex names for the graph, there are three options: (1) NA (no label) (2) NULL (default numeric label according to the feature order) (3) a vector of labels (a vector of labels cooresponding to x) deault value is NULL
type	type of graph, there are four options: (1) "task" (graph for each task (including shared part) specified further by subID (task number)) (2) "share" (shared graph for all tasks) (3) "taskspecific" (graph for each task specific (excluding shared part) specified further by subID (task number)) (4) "neighbour" (zoom into nodes in the graph specified further by neighbouroptoin, subID (task number) and index (node id))
neighbouroption	determines what type of graph to zoom into when parameter type is "neighbour" There are two options: (1) "task" (zoom into graph for each task (including shared part)) (2) "taskspecific" (zoom into graph for each task specific (excluding shared part))
subID	selects which task to display (1) 0 (only allowed when type is task or type is neighbour and neighbouroption is task) (selecting share graph) (2) positive task number (selects a task number) (3) a vector of task number (selects multiple tasks) (4) NULL (selects all tasks (all graphs))
index	determines which node(s) to zoom into when parameter type is "neighbour" could either be an integer or vector of integers representing node ids (zoom into one node or multiple nodes)
graphlayout	layout for the graph (two column matrix specifying x,y coordinates of each node in graph) if not provided, igraph will use the default layout_nicely() function present the graph
...	extra parameters passed to plot.igraph

Details

when only the simulresult is provided, the function will plot all graphs with default numeric labels User can specify multiple subID and multiple index to zoom in multiple nodes on multiple graphs Each graph will include a decriptive title and legend to indicate correspondence between edge color and task. The function will plot graph and return an igraph object at the same time

Value

a plot of graph / subgraph from simule result specified by user input

Author(s)

Beilun Wang, Zhaoyang Wang (Author), Beilun Wang (maintainer)

Examples

```
## Not run:
data(exampleData)
```

```
result = simule(X = exampleData , lambda = 0.1, epsilon = 0.45, covType = "cov", FALSE)
plot.simule(result,  graphlabel = NULL, type="task", graphlayout = NULL)

## End(Not run)
```

`returngraph` *return igraph object from simule result specified by user input*

Description

This function can return an igraph object from simule result for user to work with directly

Usage

```
returngraph(x, type = "task", neighbouroption = "task", subID = NULL,
  index = NULL)
```

Arguments

`x`	output generated from simule/wsimule function (simule/wsimule class)
`type`	type of graph, there are four options: (1) "task" (graph for each task (including shared part) specified further by subID (task number)) (2) "share" (shared graph for all tasks) (3) "taskspecific" (graph for each task specific (excluding shared part) specified further by subID (task number)) (4) "neighbour" (zoom into nodes in the graph specified further by neighbouroptoin, subID (task number) and index (node id))
`neighbouroption`	determines what type of graph to zoom into when parameter type is "neighbour" There are two options: (1) "task" (zoom into graph for each task (including shared part)) (2) "taskspecific" (zoom into graph for each task specific (excluding shared part))
`subID`	selects which task to display (1) 0 (only allowed when type is task or type is neighbour and neighbouroption is task) (selecting share graph) (2) positive task number (selects a task number) (3) a vector of task number (selects multiple tasks) (4) NULL (selects all tasks (all graphs))
`index`	determines which node(s) to zoom into when parameter type is "neighbour" could either be an integer or vector of integers representing node ids (zoom into one node or multiple nodes)

Details

the function aims to provide users the flexibility to explore and visualize the graph own their own generated from simule / wsimule

Value

an igraph object of graph / subgraph from simule result specified by user input

Author(s)

Beilun Wang, Zhaoyang Wang (Author), Beilun Wang (maintainer)

Examples

```
## Not run:
data(exampleData)
result = simule(X = exampleData , lambda = 0.1, epsilon = 0.45, covType = "cov", TRUE)
graph = returngraph(result, type="task")

## End(Not run)
```

`simule` *A constrained l1 minimization approach for estimating multiple Sparse Gaussian or Nonparanormal Graphical Models Estimate multiple, related sparse Gaussian or Nonparanormal graphical*

Description

models from multiple related datasets using the SIMULE algorithm. Please run demo(simule) to learn the basic functions provided by this package. For further details, please read the original paper: Beilun Wang, Ritambhara Singh, Yanjun Qi (2017) <DOI:10.1007/s10994-017-5635-7>.

Usage

```
simule(X, lambda, epsilon = 1, covType = "cov", parallel = FALSE)
```

Arguments

`X`	A List of input matrices. They can be data matrices or covariance/correlation matrices. If every matrix in the X is a symmetric matrix, the matrices are assumed to be covariance/correlation matrices. More details at <https://github.com/QData/SIMULE>
`lambda`	A positive number. The hyperparameter controls the sparsity level of the matrices. The \eqn\lambda_n in the following section: Details.
`epsilon`	A positive number. The hyperparameter controls the differences between the shared pattern among graphs and the individual part of each graph. The \eqn\epsilon in the following section: Details. If epsilon becomes larger, the generated graphs will be more similar to each other. The default value is 1, which means that we set the same weights to the shared pattern among graphs and the individual part of each graph.
`covType`	A parameter to decide which Graphical model we choose to estimate from the input data. If covType = "cov", it means that we estimate multiple sparse Gaussian Graphical models. This option assumes that we calculate (when input X represents data directly) or use (when X elements are symmetric representing covariance matrices) the sample covariance matrices as input to the simule algorithm.

If covType = "kendall", it means that we estimate multiple nonparanormal Graphical models. This option assumes that we calculate (when input X represents data directly) or use (when X elements are symmetric representing correlation matrices) the kendall's tau correlation matrices as input to the simule algorithm.

`parallel` A boolean. This parameter decides if the package will use the multithreading architecture or not.

Details

The SIMULE algorithm is a constrained l1 minimization method that can detect both the shared and the task-specific parts of multiple graphs explicitly from data (through jointly estimating multiple sparse Gaussian graphical models or Nonparanormal graphical models). It solves the following equation: \deqn \hat\Omega^(1)_I, \hat\Omega^(2)_I, \dots, \hat\Omega^(K)_I, \hat\Omega_S = \min\limits_\Omega^(i)_I,\Omega_S\sum\limits_i ||\Omega^(i)_I||_1+ \epsilon K||\Omega_S||_1 Subject to : \deqn ||\Sigma^(i)(\Omega^(i)_I + \Omega_S) - I||_\infty \le \lambda_n, i = 1,\dots,K \nonumber Please also see the equation (7) in our paper. The \eqn\lambda_n is the hyperparameter controlling the sparsity level of the matrices and it is the \codelambda in our function. The \eqn\epsilon is the hyperparameter controlling the differences between the shared pattern among graphs and the individual part of each graph. It is the \codeepsilon parameter in our function and the default value is 1. For further details, please see our paper: <http://link.springer.com/article/10.1007/s10994-017-5635-7>.

Value

\itemGraphsA list of the estimated inverse covariance/correlation matrices. \itemshareThe share graph among multiple tasks.

Author(s)

Beilun Wang

References

Beilun Wang, Ritambhara Singh, Yanjun Qi (2017). A constrained L1 minimization approach for estimating multiple Sparse Gaussian or Nonparanormal Graphical Models. http://link.springer.com/article/10.1007/s10994-017-5635-7

Examples

```
## Not run:
data(exampleData)
result = simule(X = exampleData , lambda = 0.1, epsilon = 0.45, covType = "cov", FALSE)
plot.simule(result)

## End(Not run)
```

wsimule *A constrained and weighted l1 minimization approach for estimating multiple Sparse Gaussian or Nonparanormal Graphical Models*

Description

Estimate multiple, related sparse Gaussian or Nonparanormal graphical models from multiple related datasets using the SIMULE algorithm. Please run demo(wsimule) to learn the basic functions provided by this package. For further details, please read the original paper: Beilun Wang, Ritambhara Singh, Yanjun Qi (2017) <DOI:10.1007/s10994-017-5635-7>.

Usage

```
wsimule(X, lambda, epsilon = 1, W, covType = "cov", parallel = FALSE)
```

Arguments

X	A List of input matrices. They can be data matrices or covariance/correlation matrices. If every matrix in the X is a symmetric matrix, the matrices are assumed to be covariance/correlation matrices. More details at <https://github.com/QData/SIMULE>
lambda	A positive number. The hyperparameter controls the sparsity level of the matrices. The λ_n in the following section: Details.
epsilon	A positive number. The hyperparameter controls the differences between the shared pattern among graphs and the individual part of each graph. The ϵ in the following section: Details. If epsilon becomes larger, the generated graphs will be more similar to each other. The default value is 1, which means that we set the same weights to the shared pattern among graphs and the individual part of each graph.
W	A weight matrix. This matrix uses the prior knowledge of the graphs. For example, if we use wsimule to infer multiple human brain connectome graphs, the W can be the anatomical distance matrix of human brain. The default value is a matrix, whose entries all equals to 1. This means that we do not have any prior knowledge.
covType	A parameter to decide which Graphical model we choose to estimate from the input data. If covType = "cov", it means that we estimate multiple sparse Gaussian Graphical models. This option assumes that we calculate (when input X represents data directly) or use (when X elements are symmetric representing covariance matrices) the sample covariance matrices as input to the simule algorithm. If covType = "kendall", it means that we estimate multiple nonparanormal Graphical models. This option assumes that we calculate (when input X represents data directly) or use (when X elements are symmetric representing correlation matrices) the kendall's tau correlation matrices as input to the simule algorithm.
parallel	A boolean. This parameter decides if the package will use the multithreading architecture or not.

Details

The SIMULE algorithm is a constrained l1 minimization method that can detect both the shared and the task-specific parts of multiple graphs explicitly from data (through jointly estimating multiple sparse Gaussian graphical models or Nonparanormal graphical models). It solves the following equation:

$$\hat{\Omega}_I^{(1)}, \hat{\Omega}_I^{(2)}, \ldots, \hat{\Omega}_I^{(K)}, \hat{\Omega}_S = \min_{\Omega_I^{(i)}, \Omega_S} \sum_i ||W \cdot \Omega_I^{(i)}||_1 + \epsilon K ||W \cdot \Omega_S||_1$$

Subject to :

$$||\Sigma^{(i)}(\Omega_I^{(i)} + \Omega_S) - I||_\infty \leq \lambda_n, i = 1, \ldots, K$$

Please also see the equation (7) in our paper. The λ_n is the hyperparameter controlling the sparsity level of the matrices and it is the `lambda` in our function. The ϵ is the hyperparameter controlling the differences between the shared pattern among graphs and the individual part of each graph. It is the `epsilon` parameter in our function and the default value is 1. For further details, please see our paper: <http://link.springer.com/article/10.1007/s10994-017-5635-7>.

Value

`Graphs`	A list of the estimated inverse covariance/correlation matrices.
`share`	The share graph among multiple tasks.

Author(s)

Beilun Wang

References

Beilun Wang, Ritambhara Singh, Yanjun Qi (2017). A constrained L1 minimization approach for estimating multiple Sparse Gaussian or Nonparanormal Graphical Models. http://link.springer.com/article/10.1007/s10994-017-5635-7

Examples

```
## Not run:
data(exampleData)
result = wsimule(X = exampleData , lambda = 0.1, epsilon = 0.45,
W = matrix(1,20,20), covType = "cov", FALSE)
plot.simule(result)

## End(Not run)
```

Index