Package 'simule'

July 31, 2017

Version 1.1.2

Date 2017-07-31

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

Author Beilun Wang [aut, cre], Yanjun Qi [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 2017-07-31 21:47:03 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:	1.1.2
Date:	2017-07-31
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

Maintainer: Beilun Wang - bw4mw at virginia dot edu

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)

## End(Not run)
```

---

cancer *Microarray data set for breast cancer*

Description

This gene expression data set is freely available, coming from the Hess *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.

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.

Usage

```
data(cancer)
```

Format

A list named `cancer` comprising two objects:

`expr` a `data.frame` with 26 columns and 133 rows. The nth row gives the expression levels of the 26 identified genes for the nth patient. The columns are named according to the genes.

`status` a factor of size 133 with 2 levels (`"pcr"` and `"not"`), describing the status of the patient.

References

K.R. Hess, K. Anderson, W.F. Symmans, V. Valero, N. Ibrahim, 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.

Examples

```
## load the breast cancer data set
data(cancer)
attach(cancer)
```

`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(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>.

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

Examples

```
data(exampleData)
```

net.degree *List the degree of every node of each graph in the input list of multiple graphs.*

Description

Lists the degree of every node of each graph in the input list of multiple graphs.

Usage

```
net.degree(theta)
```

Arguments

theta	An input list of multiple graphs. Each graph is represented as a pXp matrix. (For example, the result of the SIMULE algorithm: a list of pXp matrices in which each matrix represents an estimated sparse inverse covariance matrix.)

Value

Degrees, in the format of a list of length p vectors represents the degree of all p nodes of each graph in the input list of multiple graphs.

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:
## load an examplar multi-task dataset with K=2 tasks, p=100 features, and n=200 samples per task:
data(exampleData)
##run simule
result = simule(X = exampleData , 0.05, 1, covType = "cov", FALSE)
## get degree list:
net.degree(result$Graphs)

## End(Not run)
```

net.edges *List the edges of each graph in the input list of multiple graphs*

Description

List every estimated edge in the form of pair of connected nodes for each graph in the input list of multiple graphs.

Usage

```
net.edges(theta)
```

Arguments

theta	An input list of multiple graphs. Each graph is represented as a pXp matrix. (For example, the result of the SIMULE algorithm: a list of pXp matrices in which each matrix represents an estimated sparse inverse covariance matrix.)

Value

edges, a length K list, each element of the list represents an igraph.es object which is the detail of all pairs of connected nodes of each graph in the input list of multiple graphs.

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:
## load an example multi-task dataset with K=2 tasks, p=100 features, and n=200 samples per task:
data(exampleData)
##run simule
result = simule(X = exampleData , 0.05, 1, covType = "cov", FALSE)
## get edges list:
net.edges(result$Graphs)

## End(Not run)
```

net.hubs *Get degrees of the most connected nodes of each graph in the input list of multiple graphs.*

Description

List the degrees of the hub nodes of each graph in the input list of multiple graphs.

Usage

```
net.hubs(theta, nhubs = 10)
```

Arguments

theta	An input list of multiple graphs. Each graph is represented as a pXp matrix. (For example, the result of the SIMULE algorithm: a list of pXp matrices in which each matrix represents an estimated sparse inverse covariance matrix.)
nhubs	The number of hubs to be identified of each graph in the input list of multiple graphs.

Value

hubs, a length K list. Each element in this list is a vector of length nhubs whose entries give the degree of the most connected nodes of each graph in the input list of multiple graphs.

Author(s)

Beilun Wang

References

Beilun Wang, Ritambhara Singh and Yanjun Qi (2017). A Constrained L1 Minimization Approach for Estimating Multiple Sparse Gaussian or Nonparanormal Graphical Models. <http://link.springer.com/article/10.1007/s109 017-5635-7>

Examples

```
## Not run:
## load an example multi-task dataset with K=2 tasks, p=100 features, and n=200 samples per task:
data(exampleData)
##run simule
result = simule(X = exampleData , 0.05, 1, covType = "cov", FALSE)
## get hubs list:
net.hubs(result$Graphs)

## End(Not run)
```

net.neighbors *Get neighbors of a node in each graph in the input list of multiple graphs*

Description

For each graph in the input list of multiple graphs, returns the name of neighbor nodes connected to a given node.

Usage

```
net.neighbors(theta, index)
```

Arguments

theta	An input list of multiple graphs. Each graph is represented as a pXp matrix. (For example, the result of the SIMULE algorithm: a list of pXp matrices in which each matrix represents an estimated sparse inverse covariance matrix.)
index	The row number of the node to be investigated.

Value

neighbors, a length K list. Each element in the list is a vector including row names of the neighbor nodes for the index node in each graph in the input list of multiple graphs.

Author(s)

Beilun Wang

References

Beilun Wang, Ritambhara Singh and Yanjun Qi (2017). A Constrained L1 Minimization Approach for Estimating Multiple Sparse Gaussian or Nonparanormal Graphical Models. <http://link.springer.com/article/10.1007/s109 017-5635-7>

Examples

```
## Not run:
## load an example multi-task dataset with K=2 tasks, p=100 features, and n=200 samples per task:
data(exampleData)
##run simule
result = simule(X = exampleData , 0.05, 1, covType = "cov", FALSE)
## get neighbors of node 50:
net.neighbors(result$Graphs,index=50)

## End(Not run)
```

plot.simule *Plotting functions for displaying the list of multiple graphs generated by the simule algorithm*

Description

This function plots either the shared graph, the task-specific networks, the networks or the neighborhood networks for a certain node. 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>.

Usage

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

Arguments

x	simule object
type	Plotting type. This argument defines which type of network(s) to plot. There are four options: "graph": plot the networks. The different colors represent the different graphs. "share": plot the shared graph. "sub": plot subject-specific networks. "neighbor": plot the neighborhood networks for a given node. The different colors represent the different graphs.
subID	If type="sub", subID indicates to plot the task-specific network for the task whose index == subID.
index	If type="neighbor", index indicates the row number of the node to be investigated. This function plots its neighborhood networks in each graph of the multiple graphs generated by simule algorithm.
...	Additional arguments to pass to plot function

Details

Plotting function for simule objects. It can be used to plot results obtained from running the simule algorithm.

Author(s)

Beilun Wang and Yanjun Qi

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

See Also

simule

Examples

```
## Not run:
  data(exampleData)
  results = simule(X = exampleData , 0.05, 1, covType = "cov", TRUE)
  plot.simule(results)
  plot.simule(results, type="share")
  plot.simule(results, type="sub", subID=1)
  plot.simule(results, type="neighbor", index=50)

## End(Not run)
```

---

simule *A constrained 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(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>.

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 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.
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:

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

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)
results = simule(X = exampleData , 0.05, 1, covType = "cov", TRUE)
plot.simule(results)
plot.simule(results, type="share")
plot.simule(results, type="sub", subID=1)
```

```
plot.simule(results, type="neighbor", index=50)

## End(Not run)
```

Index