R documentation

of 'simule-package.Rd' etc.

July 4, 2017

---

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. It also provides two computational options: the multi-threading implementation and the single-threading implementation.

Details

Package:	simule
Type:	Package
Version:	1.1
Date:	2017-05-08
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/s1099
017-5635-7

Examples

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

## End(Not run)
```

`simule` *A constrained l1 minimization approach for estimating multiple Sparse Gaussian or Nonparanormal Graphical Models*

Description

Estimate multiple, related graphical models across multiple related datasets using the SIMULE algorithm.

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.
`lambda`	A positive number. The hyperparameter controls the sparsity level of the matrices. The λ in the 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 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. If covType = "cov", it means that we estimate multiple Gaussian Graphical models. In another word, we assume that the $\Sigma^{(i)}$ in the details are estimated by the sample covariance matrices. If covType = "cor", it means that we estimate multiple nonparanormal Graphical models. In another word, we assume that the $\Sigma^{(i)}$ in the details are estimated by the kendall's tau correlation matrices.

`parallel` A boolean. It is the parameter to decide that the package uses the multithreading architecture or not.

Details

The SIMULE algorithm is a constrained l1 minimization method to detect shared and individual parts of multiple graphs explicitly.

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

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 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 matrices.

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/s1099
017-5635-7

Examples

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

plot.simule *Plotting function for simule objects*

Description

Plotting function for simule objects. This function implements plotting for either the original multiple graphs, the shared graph, the subject-specific networks, or the neighborhood networks for certain node.

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) is chosen by the user. There are four options: "graph": plot the original networks, "share": plot the shared graph, "sub": plot subject-specific network, and "neighbor": plot the neighborhood networks for a given node.
subID	If type="sub", subID indicates which subject-specific network should be plotted.
index	If type="neighbor", index indicates which node to choose for plotting the neighborhood networks.
...	Additional arguments to pass to plot function

Details

Plotting function for simule objects. It can be used to plot simulated networks or results obtained from running 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/s1099
017-5635-7

See Also

simule

Examples

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

`net.hubs` *Get degrees of most connected nodes.*

Description

List the degrees of the most connected nodes in each class.

Usage

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

Arguments

`theta`	A list of pXp matrices, each element represents an estimated sparse inverse covariance matrix. (For example, the result of SIMULE.)
`nhubs`	The number of hubs to be identified.

Value

hubs, a list of length K, each element is a vector of Length n hubs giving the degree of the most connected nodes of the corresponding class.

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. <arXiv:1605.03468>

Examples

```
## Not run:
## load an example 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 the network*

Description

For each class, returns the name of neighbour nodes connected to a given node.

Usage

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

Arguments

theta	A list of pXp matrices, each element represents an estimated sparse inverse covariance matrix. (For example, the result of SIMULE.)
index	The row number of the node to be investigated.

Value

neighbors, a list of length K, each element is a vector including row names of the neighbour nodes.

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. <arXiv:1605.03468>

Examples

```
## Not run:
## load an example 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)
```

net.degree *List the degree of every node in all classes.*

Description

Lists the degree of every node for each class.

Usage

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

Arguments

theta	A list of pXp matrices, each element represents an estimated sparse inverse covariance matrix. (For example, the result of the SIMULE algorithm.)

Value

Degrees, in the format of a list of length p vectors represents the degree of all p nodes in the network for a class.

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. <arXiv:1605.03468>

Examples

```
## Not run:
## load an example 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 in a network*

Description

For each class, list every edge in the form of pair of connected nodes.

Usage

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

Arguments

theta	A list of pXp matrices, each element of the list represents an estimated sparse inverse covariance matrix. (For example, the result of SIMULE.)

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 in the class.

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. <arXiv:1605.03468>

Examples

```
## Not run:
## load an example 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)
```

Index