Package 'jeek'

June 14, 2018

Type Package

Date 2018-06-15

Title A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models

Version 1.0.0

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

Maintainer Beilun Wang <bw4mw@virginia.edu>

Description
Provides a fast and scalable joint estimator for integrating additional knowledge in learning multiple related sparse Gaussian Graphical Models (JEEK). The JEEK algorithm can be used to fast estimate multiple related precision matrices in a large-scale. For instance, it can identify multiple gene networks from multi-context gene expression datasets. By performing data-driven network inference from high-dimensional and heterogeneous data sets, this tool can help users effectively translate aggregated data into knowledge that take the form of graphs among entities. Please run demo(jeekDemo) to learn the basic functions provided by this package. For further details, please read the original paper: Beilun Wang, Arshdeep Sekhon, Yanjun Qi ``A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models" (ICML 2018) <arXiv:1806.00548>.

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

Suggests parallel

License GPL-2

Encoding UTF-8

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

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

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-06-14 19:24:50 UTC

R topics documented:

`jeek-package` *A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models*

Description

This is an R implementation of a Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models (JEEK).The JEEK algorithm can be used to fast estimate multiple related precision matrices in a large-scale. For instance, it can identify multiple 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(jeekDemo) to learn the basic functions provided by this package. For further details, please read the original paper: Beilun Wang, Arshdeep Sekhon, Yanjun Qi (2018).

Details

Package:	jeek
Type:	Package
Version:	1.0.0
Date:	2018-06-15
License:	GPL (>= 2)

We consider the problem of including additional knowledge in estimating sparse Gaussian graphical models (sGGMs) from aggregated samples, arising often in bioinformatics and neuroimaging applications. Previous joint sGGM estimators either fail to use existing knowledge or cannot scale-up to many tasks (large K) under a high-dimensional (large p) situation. In this paper, we propose a novel Joint Elementary Estimator incorporating additional Knowledge (JEEK) to infer multiple related sparse Gaussian Graphical models from large-scale heterogeneous data. Using domain knowledge as weights, we design a novel hybrid norm as the minimization objective to enforce the superposition of two weighted sparsity constraints, one on the shared interactions and the other on the

task-specific structural patterns. This enables JEEK to elegantly consider various forms of existing knowledge based on the domain at hand and avoid the need to design knowledge-specific optimization. JEEK is solved through a fast and entry-wise parallelizable solution that largely improves the computational efficiency of the state-of-the-art $O(p^5K^4)$ to $O(p^2K^4)$. We conduct a rigorous statistical analysis showing that JEEK achieves the same convergence rate $O(\log(Kp)/n_{tot})$ as the state-of-the-art estimators that are much harder to compute. Empirically, on multiple synthetic datasets and one real-world data from neuroscience, JEEK outperforms the speed of the state-of-arts significantly while achieving the same level of prediction accuracy.

Author(s)

Beilun Wang

Maintainer: Beilun Wang - bw4mw at virginia dot edu

References

Beilun Wang, Arshdeep Sekhon, Yanjun Qi. A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models. <arXiv:1806.00548>

Examples

```
data(cancer)
X = list(as.matrix(cancer[[1]][which(cancer[[2]] == "not"),]),
as.matrix(cancer[[1]][which(cancer[[2]] == "pcr"),]))
jeek(X, 0.05, covType = "cov", parallel = FALSE)
```

---

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(jeekDemo) 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)
```

jeek *A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models*

Description

A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models. Please run demo(jeekDemo) to learn the basic functions provided by this package. For further details, please read the original paper: Beilun Wang, Arshdeep Sekhon, Yanjun Qi (2018).

Usage

```
jeek(X, lambda, 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/JEEK>
lambda	A positive number. The hyperparameter controls the sparsity level of the matrices. The λ_n in the following section: Details.
W	A list of weight matrices. The hyperparameter intergrating the additional knowledge into the model. The W_{ij} is large means that node i and node j have less probability to connect with each other. The default value of each entry is 1, which means there is no additional knowledge in the formulation.
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 JEEK 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 JEEK algorithm.
parallel	A boolean. This parameter decides if the package will use the multithreading architecture or not.

Details

The JEEK algorithm is a novel Joint Elementary Estimator incorporating additional Knowledge (JEEK) to infer multiple related sparse Gaussian Graphical models from large-scale heterogeneous data. It solves the following equation:

$$\min_{\Omega_I^{tot},\Omega_S^{tot}} ||W_I^{tot} \circ \Omega_I^{tot}||_1 + ||W_S^{tot} \circ \Omega_S^{tot}||$$

Subject to :

$$||W_I^{tot} \circ (\Omega^{tot} - inv(T_v(\hat{\Sigma}^{tot})))||_\infty \leq \lambda_n$$

$$||W_S^{tot} \circ (\Omega^{tot} - inv(T_v(\hat{\Sigma}^{tot})))||_\infty \leq \lambda_n$$

$$\Omega^{tot} = \Omega_S^{tot} + \Omega_I^{tot}$$

Please also see the equation (3.7) in our paper. The λ_n is the hyperparameter controlling the sparsity level of the matrices and it is the `lambda` in our function. For further details, please see our paper: Beilun Wang, Arshdeep Sekhon, Yanjun Qi. A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models. ICML 2018

Value

`Graphs` A list of the estimated inverse covariance/correlation matrices.

Author(s)

Beilun Wang

References

Beilun Wang, Arshdeep Sekhon, Yanjun Qi. A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models. <arXiv:1806.00548>

Examples

```
data(cancer)
X = list(as.matrix(cancer[[1]][which(cancer[[2]] == "not"),]),
as.matrix(cancer[[1]][which(cancer[[2]] == "pcr"),]))
results = jeek(X, 0.05, covType = "cov", parallel = FALSE)
plot.jeek(results)
plot.jeek(results, type="neighbor", index=10)
```

`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, Arshdeep Sekhon, Yanjun Qi. A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models. ICML 2018

Examples

```
data(cancer)
X = list(as.matrix(cancer[[1]][which(cancer[[2]] == "not"),]),
as.matrix(cancer[[1]][which(cancer[[2]] == "pcr"),]))
##run jeek
result = jeek(X, 0.05, covType = "cov", parallel = FALSE)
## get degree list:
net.degree(result$Graphs)
```

---

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, Arshdeep Sekhon, Yanjun Qi. A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models. ICML 2018

Examples

```
data(cancer)
X = list(as.matrix(cancer[[1]][which(cancer[[2]] == "not"),]),
as.matrix(cancer[[1]][which(cancer[[2]] == "pcr"),]))
##run jeek
result = jeek(X, 0.05, covType = "cov", parallel = FALSE)
## get edges list:
net.edges(result$Graphs)
```

---

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, Arshdeep Sekhon, Yanjun Qi. A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models. ICML 2018

Examples

```
data(cancer)
X = list(as.matrix(cancer[[1]][which(cancer[[2]] == "not"),]),
as.matrix(cancer[[1]][which(cancer[[2]] == "pcr"),]))
##run jeek
result = jeek(X, 0.05, covType = "cov", parallel = FALSE)
## get hubs list:
net.hubs(result$Graphs)
```

---

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, Arshdeep Sekhon, Yanjun Qi. A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models. ICML 2018

Examples

```
data(cancer)
X = list(as.matrix(cancer[[1]][which(cancer[[2]] == "not"),]),
as.matrix(cancer[[1]][which(cancer[[2]] == "pcr"),]))
##run jeek
result = jeek(X, 0.05, covType = "cov", parallel = FALSE)
## get neighbors of node 10:
net.neighbors(result$Graphs,index=10)
```

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

Description

This function plots the networks or the neighborhood networks for a certain node. Please run demo(jeekDemo) to learn the basic functions provided by this package. For further details, please read the original paper: Beilun Wang, Arshdeep Sekhon, Yanjun Qi (2018).

Usage

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

Arguments

`x`	jeek 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. "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 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 jeek algorithm.
`...`	Additional arguments to pass to plot function

Details

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

Author(s)

Beilun Wang and Yanjun Qi

References

Beilun Wang, Arshdeep Sekhon, Yanjun Qi. A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models. ICML 2018

See Also

jeek

Examples

```
data(cancer)
X = list(as.matrix(cancer[[1]][which(cancer[[2]] == "not"),]),
as.matrix(cancer[[1]][which(cancer[[2]] == "pcr"),]))
results = jeek(X, 0.05, covType = "cov", parallel = FALSE)
plot.jeek(results)
plot.jeek(results, type="neighbor", index=10)
```

Index