Package 'jeek'

July 7, 2018

Type Package

Date 2018-07-03

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

Version 1.1.1

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(jeek) 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

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

Repository CRAN

Date/Publication 2018-07-07 15:40:13 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(jeek) 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:	1.1.0
Date:	2018-07-03
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, Zhaoyang 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

```
## Not run:
data(exampleData)
result = jeek(X = exampleData, 0.3, covType = "cov", parallel = TRUE)
plot.jeek(results)

## 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

`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(jeek) 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 = NA, 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

```
## Not run:
data(exampleData)
result = jeek(X = exampleData, 0.3, covType = "cov", parallel = TRUE)
plot.jeek(results)

## End(Not run)
```

`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.jeek` *Plot jeek result specified by user input*

Description

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

Usage

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

Arguments

`x`	output generated from the "jeek" function (jeek 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 jeek result specified by user input

Author(s)

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

Examples

```
## Not run:
data(exampleData)
result = jeek(X = exampleData, 0.3, covType = "cov", parallel = TRUE)
plot.jeek(result)

## End(Not run)
```

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

Description

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

Usage

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

Arguments

`x`	output generated from jeek function (jeek 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 on their own generated from jeek

Value

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

Author(s)

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

Examples

```
## Not run:
data(exampleData)
result = jeek(X = exampleData, 0.3, covType = "cov", parallel = TRUE)
graph = returngraph(result)

## End(Not run)
```

Index