Package 'fasjem'

October 13, 2022

Version 1.1.2

Date 2017-07-31

Title A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models

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

Maintainer Beilun Wang <bw4mw@virginia.edu>

Depends R (>= 3.0.0), igraph

Description

This is an R implementation of ``A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models" (FASJEM). The FASJEM 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 heterogonous data sets, this tool can help users effectively translate aggregated data into knowledge that take the form of graphs among entities. Please run demo(fasjem) to learn the basic functions provided by this package. For more details, please see <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>.

License GPL-2

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

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

NeedsCompilation no

Repository CRAN

Date/Publication 2017-08-01 05:05:24 UTC

R topics documented:

`fasjem-package` *A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models*

Description

This is an R implementation of "A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models" (FASJEM). The FASJEM 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. For more details, please read <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>. Assuming the multiple related graphs share a certain sparsity pattern, this package provides two different options for regularizing such sparsity patterns: (1) the group,2 norm (`method = "fasjem-g"`) and (2) the group,infinity norm (`method = "fasjem-i"`).

Details

Package:	fasjem
Type:	Package
Version:	1.1.2
Date:	2017-07-31
License:	GPL (>= 2)

Estimating multiple sparse Gaussian Graphical Models (sGGMs) jointly for many related tasks (large K) under a high-dimensional (large p) situation is an important task. Most previous studies for the joint estimation of multiple sGGMs rely on penalized log-likelihood estimators that involve expensive and difficult non-smooth optimizations. We propose a novel approach, FASJEM for FAst and Scalable Joint structure-Estimation of Multiple sGGMs at a large scale. As the first study of joint sGGM using the M-estimator framework, our work has three major contributions: (1) We solve FASJEM through an entry-wise manner which is parallelizable. (2) We choose a proximal algorithm to optimize FASJEM. This improves the computational efficiency from $O(Kp^3)$ to $O(Kp^2)$ and reduces the memory requirement from $O(Kp^2)$ to $O(K)$. (3) We theoretically prove that FASJEM achieves a consistent estimation with a convergence rate of $O(log(Kp)/n_{tot})$. On several synthetic and four real-world datasets, FASJEM shows significant improvements over baselines on accuracy, computational complexity and memory costs.

Author(s)

Beilun Wang

Maintainer: Beilun Wang - bw4mw at virginia dot edu

References

Beilun Wang, Ji Gao, Yanjun Qi (2017). A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models. <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>

Examples

```
## Not run:
data(exampleData)
fasjem(X = exampleData, method = "fasjem-g", 0.1, 0.1, 0.1, 0.05, 10)
demo(fasjem)

## End(Not run)
```

`exampleData` *A simulated toy dataset that includes 2 data matrices (about 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(fasjem) to learn the basic functions provided by this package. For further details, please read the original paper: <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>.

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)
```

`fasjem` *A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models*

Description

The R implementation of the FASJEM method, which is introduced in the paper "A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models". Please run demo(fasjem) to learn the basic functions provided by this package. For more details, please see <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>.

Usage

```
fasjem(X, method="fasjem-g", lambda=0.1, epsilon=0.1, gamma=0.1, rho=0.05, iterMax=10)
```

Arguments

`X`	A List of input matrices. They can be either data matrices or covariance matrices. If every matrix in the X is a symmetric matrix, the input matrices are assumed to be the covariance matrices from the multiple related tasks.
`method`	By using two different regularization functions as the second norm in the objective, this package provides two different options for regularizing the sparsity pattern shared among multiple graphs. This parameter decides which function to use for the second regularization norm. When `method = "fasjem-g"`, fasjem will use the group,2 norm as the second regularization function. When `method = "fasjem-i"`, fasjem will choose the group,infinity norm as the second regularization function. The default value is "fasjem-g". Please check the paper for more details.
`lambda`	A positive number. This hyperparameter controls the sparsity level of the matrices. The λ_n in the following section: Details.
`epsilon`	A positive number. This hyperparameter represents the ratio between the l1 norm and the second group norm. The ϵ in the following section: Details.
`gamma`	A positive number. This hyperparameter is used in calculating each proximity during optimization. Please check the Algorithm 1 in our paper for more details.
`rho`	A positive number. This hyperparameter controls the learning rate of the proximal gradient method. Please check the Algorithm 1 in our paper for more details.
`iterMax`	An integer. The max number of iterations in the optimization of fasjem.

Details

The FASJEM algorithm is a fast and scalable method to estimate multiple related sparse Gaussian Graphical models. It solves the following equation:

$$\min_{\Omega_{tot}} ||\Omega_{tot}||_1 + \epsilon \mathcal{R}'(\Omega_{tot})$$

Subject to :

$$||\Omega_{tot} - inv(T_v(\hat{\Sigma}_{tot}))||_\infty \leq \lambda_n$$

$$\mathcal{R}'^*(\Omega_{tot} - inv(T_v(\hat{\Sigma}_{tot}))) \leq \epsilon\lambda_n$$

More details are provided in the equation (3.1) of our original paper.

The λ_n in the above equation represents the hyperparameter `lambda` who controls the sparsity level of the target precision matrices.

The $\epsilon\lambda_n$ in the above equation represents the regularization parameter of the second norm who controls how multiple graphs share a certain pattern. Here ϵ represents the input parameter `epsilon` whose default value is 0.1.

Other parameters in the fasjem function are described in details by the Algorithm 1 in our paper.

When `method = "fasjem-g"`, $\mathcal{R}'(\cdot) = ||\cdot||_{\mathcal{G},2}$.

When `method = "fasjem-i"`, $\mathcal{R}'(\cdot) = ||\cdot||_{\mathcal{G},\infty}$.

Please run `demo(fasjem)` to learn the basic functions provided by this package. For more details, please see <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>.

Value

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

References

Beilun Wang, Ji Gao, Yanjun Qi (2017). A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models. <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>

Examples

```
data(exampleData)
fasjem(X = exampleData, method = "fasjem-g", 0.1, 0.1, 0.1, 0.05, 10)
fasjem(X = exampleData, method = "fasjem-i", 0.1, 0.1, 0.1, 0.05, 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 fasjem 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, Ji Gao, Yanjun Qi (2017). A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models. <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>

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
result = fasjem(X = exampleData, method = "fasjem-g", 0.1, 0.1, 0.1, 0.05, 10)
## get degree list:
net.degree(result)

## 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 fasjem 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, Ji Gao, Yanjun Qi (2017). A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models. <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>

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
result = fasjem(X = exampleData, method = "fasjem-g", 0.1, 0.1, 0.1, 0.05, 10)
## get edges list:
net.edges(result)

## 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 fasjem 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, Ji Gao, Yanjun Qi (2017). A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models. <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>

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
result = fasjem(X = exampleData, method = "fasjem-g", 0.1, 0.1, 0.1, 0.05, 10)
## get hubs list:
net.hubs(result)

## 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 fasjem 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, Ji Gao, Yanjun Qi (2017). A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models. <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>

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
result = fasjem(X = exampleData, method = "fasjem-g", 0.1, 0.1, 0.1, 0.05, 10)
## get neighbors of node 50:
net.neighbors(result, index=50)

## End(Not run)
```

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

Description

Plotting function for fasjem objects. This function plots either the shared graph, the task-specific networks, the networks or the neighborhood networks for a certain node. Please run demo(fasjem) to learn the basic functions provided by this package. For further details, please read the original paper: <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>.

Usage

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

Arguments

`x`	fasjem 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 subgraphs from each graph of the multiple graphs generated by fasjem algorithm.
`...`	Additional arguments to pass to plot function

Details

Plotting function for fasjem objects. It plots the results obtained from running fasjem algorithm.

Author(s)

Beilun Wang and Yanjun Qi

References

Beilun Wang, Ji Gao, Yanjun Qi (2017). A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models. <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>

See Also

fasjem

Examples

```
  ## Not run:
    data(exampleData)
    results = fasjem(X = exampleData, method = "fasjem-g", 0.1, 0.1, 0.1, 0.05, 10)
    plot.fasjem(results)
    plot.fasjem(results, type="share")
    plot.fasjem(results, type="sub", subID=1)
    plot.fasjem(results, type="neighbor", index=50)

## End(Not run)
```

Index