R documentation

of ‘fasjem-package.Rd’ etc.

July 4, 2017

`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 heterogonous 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 see <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>. This package includes two options of the second regularization function: (1) the group,2 norm and (2) the group,infinity norm. Both of them assume that multiple graphs have similar sparisity pattern.

Details

Package:	fasjem
Type:	Package
Version:	1.1.1
Date:	2017-07-04
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)

## End(Not run)
```

`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 "A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models".

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/correlation matrices. If every matrix in the X is a symmetric matrix, the matrices are assumed to be covariance/correlation matrices you give to use.
`method`	The parameter that decides which second regularization function to use. When `method = "fasjem-g"`, the user chooses the group,2 norm as the second regularization function. When `method = "fasjem-i"`, the user chooses the group,infinity norm as the second regularization function. The default value is "fasjem-g". Please check the details for more information.
`lambda`	A postive number. It is a hyperparameter that controls the sparsity level of the matrices.
`epsilon`	A postive number. The hyperparameter represents the ratio between l1 norm and the second group norm.
`gamma`	A postive number. A hyperparameter used in calculating each proximity. Please check the Algorithm 1 in our paper for more information.
`rho`	A postive number. The learning rate of the proximal gradient method. Please check the Algorithm 1 in our paper for more information.
`iterMax`	An integer. The max number of iterations in the optimization.

Details

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

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

Subject to :

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

Please also see the equation (3.1) 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 $\epsilon\lambda_n$ represents the regularization parameter for the second norm, where ϵ is the `epsilon` parameter in our function and the default value is 0.1. Other parameters in the fasjem function are decribed in Algorithm 1 in our paper. When `method = "fasjem-g"`, it means that $\mathcal{R}'(\cdot) = ||\cdot||_{\mathcal{G},2}$. When `method = "fasjem-i"`, it means that $\mathcal{R}'(\cdot) = ||\cdot||_{\mathcal{G},\infty}$. For further details, please see our paper: <http://proceedings.mlr.press/v54/wang17e/wang17e.pdf>.

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)
```

`plot.fasjem` *Plotting function for fasjem objects*

Description

Plotting function for fasjem 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 'fasjem'
plot(x, type="graph", subID=NULL, index=NULL, ...)
```

Arguments

`x`	fasjem 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 fasjem objects. It can be used to plot simulated networks or 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)
```

---

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

Description

List the degrees of the hub 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.
nhubs	The number of hubs to be identified.

Value

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

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

```
## load an example dataset with K=2 tasks, p=100 features, and n=200 samples per task:
data(exampleData)
##run fasjem
result = fasjem(X = exampleData, method = "fasjem-g", 0.1, 0.1, 0.1, 0.05, 10)
## get hubs list:
net.hubs(result)
```

---

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 length K list. Each element in the list is a vector including row names of the neighbour nodes for the index node.

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

```
## load an example dataset with K=2 tasks, p=100 features, and n=200 samples per task:
data(exampleData)
##run fasjem
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)
```

`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 in which each matrix represents an estimated sparse inverse covariance matrix. (For example, the result of the FASJEM algorithm.)

Value

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

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

```
## load an example dataset with K=2 tasks, p=100 features, and n=200 samples per task:
data(exampleData)
##run fasjem
result = fasjem(X = exampleData, method = "fasjem-g", 0.1, 0.1, 0.1, 0.05, 10)
## get degree list:
net.degree(result)
```

`net.edges` *List the edges in a network*

Description

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

Usage

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

Arguments

`theta`	A list of pXp matrices. Each matrix in the list 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 in the class.

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

```
## load an example dataset with K=2 tasks, p=100 features, and n=200 samples per task:
data(exampleData)
##run fasjem
result = fasjem(X = exampleData, method = "fasjem-g", 0.1, 0.1, 0.1, 0.05, 10)
## get edges list:
net.edges(result)
```

Index