

Package ‘mapggm’

October 8, 2015

Type Package

Title Multi-attribute network construction and perturbation detection

Version 0.0.0

Date 2015-09-30

Author Paula J Griffin

Maintainer Paula J Griffin <paulajgriffin@gmail.com>

Description

Companion package to 'Detection of multiple perturbations in multi-omics biological networks'

License Apache License

Imports Matrix, mvtnorm, igraph

R topics documented:

getLambdaRange	1
multiAttEstimate	2
multiAttLasso	3
multiAttSelect	4
perturbNodeTests	5
perturbTests	6
seqPerturbTests	7

Index	9
--------------	----------

getLambdaRange	<i>Generate lambdas to try</i>
----------------	--------------------------------

Description

Generate penalty parameters spanning a range of 1 to `length(unique(id))` submatrix blocks. This function is provided to establish a range of reasonable penalty parameters for optimization according to the algorithm of Kolar et al (2014).

Usage

```
getLambdaRange(S, id, length.out = 10)
```

Arguments

<code>S</code>	sample covariance
<code>id</code>	vector of node identifiers
<code>length.out</code>	number of lambda parameters to return

Value

vector of `length.out` equally spaced lambdas

References

Kolar, M., Liu, H., and Xing, E. P. (2014). Graph estimation from multi-attribute data. *The Journal of Machine Learning Research* 15, 1713-1750.

Examples

```
Y <- matrix(rnorm(120), nrow=20, ncol=6)
S <- crossprod(Y)
id <- rep(1:3, each=2)
getLambdaRange(S, id, 5)
```

multiAttEstimate	<i>Wrapper function for multi-attribute network estimation via lasso</i>
------------------	--

Description

Performs estimation of a zero-mean multi-attribute Gaussian graphical model as described by Kolar et al (2014) when `mode=1`.

Usage

```
multiAttEstimate(S, n, id, lambda, W = NULL, mode = 1, update = 100,
  max.gap = 0.5, max.iter = 100, min.t = .Machine$double.eps)
```

Arguments

<code>S</code>	sample covariance matrix
<code>n</code>	number of samples
<code>id</code>	vector assigning variables to nodes
<code>lambda</code>	penalty tuning parameter
<code>W</code>	optional weight matrix
<code>mode</code>	estimation mode. Can be 1 (multiattribute; default), 2 (unstructured together), or 3 (separately; only valid for 2-attribute data)
<code>update</code>	how often to print updates in optimization
<code>max.gap</code>	maximum allowable primal/dual gap
<code>max.iter</code>	maximum number of iterations to optimize (overrides <code>max.gap</code>)
<code>min.t</code>	minimum step size (overrides <code>max.gap</code> , <code>max.iter</code>)

Details

If mode=2, the id argument is essentially disregarded, and all rows/columns in S are treated as if they represent individual nodes. If mode=3, the optimization of mode=2 is performed separately for each attribute type, and a matrix with 0 on all cross-attribute entries is returned.

Value

list of precision, covariance, optimization status, lambda, and number components

References

Kolar, M., Liu, H., and Xing, E. P. (2014). Graph estimation from multi-attribute data. The Journal of Machine Learning Research 15, 1713-1750.

Examples

```
library(mvtnorm)
id <- rep(1:3, each=2)
Omega <- matrix(0, nrow=6, ncol=6)
Omega[1:4,1:4] <- 1
diag(Omega) <- 6
Sigma <- solve(Omega)
n <- 1000
Y <- rmvnorm(n, sigma=Sigma)
S <- crossprod(Y)

lambdas <- getLambdaRange(S, id, 10)
result <- multiAttEstimate(S, n, id, lambdas[1])
```

multiAttLasso

Multi-attribute network estimation

Description

Estimates block precisions and covariances for a multi-attribute network based on a Gaussian graphical model with zero mean vector. This is according to the method described by Kolar et al (2015).

Usage

```
multiAttLasso(S, n, id, lambda, W = NULL, update = 100, max.gap = 0.5,
  max.iter = 100, min.t = .Machine$double.eps)
```

Arguments

S	sample covariance matrix
n	number of samples
id	vector of node identifiers
lambda	tuning parameter for penalty
W	optional penalty weight matrix (same dimensions as S)
update	how often to print updates
max.gap	maximum allowable primal-dual gap

<code>max.iter</code>	maximum number of iterations to complete (overrides <code>max.gap</code>)
<code>min.t</code>	minimum step size (overrides <code>max.gap</code> , <code>max.iter</code>)

Value

list of precision, covariance, optimization status, lambda, and number components

References

Kolar, M., Liu, H., and Xing, E. P. (2014). Graph estimation from multi-attribute data. The Journal of Machine Learning Research 15, 1713-1750.

<code>multiAttSelect</code>	<i>Wrapper function for multi-attribute network estimation plus selection via lasso</i>
-----------------------------	---

Description

Performs estimation of a zero-mean multi-attribute Gaussian graphical model as described by Kolar et al (2014) when `mode=1` with a selection procedure to determine optimal lambda. Selection may be performed according to extended BIC, BIC, AIC, or Hamming distance to true graph. The true graph Theta is only required if Hamming distance is being used for the selection procedure.

Usage

```
multiAttSelect(S, n, id, lambda.range, W = NULL, mode = 1, update = 100,
  max.gap = 0.5, max.iter = 100, min.t = .Machine$double.eps,
  method = "BIC", Theta.true = NULL, plot = NULL, gamma = 0.5)
```

Arguments

<code>S</code>	sample covariance matrix
<code>n</code>	number of samples
<code>id</code>	vector assigning variables to nodes
<code>lambda.range</code>	vector of lambdas to try
<code>W</code>	optional weight matrix
<code>mode</code>	estimation mode. Can be 1 (multiattribute; default), 2 (unstructured together), or 3 (separately; only valid for 2-attribute data)
<code>update</code>	how often to print updates in optimization
<code>max.gap</code>	maximum allowable primal/dual gap
<code>max.iter</code>	maximum number of iterations to optimize (overrides <code>max.gap</code>)
<code>min.t</code>	minimum step size (overrides <code>max.gap</code> , <code>max.iter</code>)
<code>method</code>	how to select the best model (EBIC, BIC, AIC, or hamming)
<code>Theta.true</code>	true underlying graph (hamming method only)
<code>plot</code>	boolean, whether or not to make diagnostic plot
<code>gamma</code>	gamma parameter for EBIC method (default 0.5)

Details

If mode=2, the id argument is essentially disregarded, and all rows/columns in S are treated as if they represent individual nodes. If mode=3, the optimization of mode=2 is performed separately for each attribute type, and a matrix with 0 on all cross-attribute entries is returned.

Value

list of precision, covariance, optimization status, lambda, and number components

References

Kolar, M., Liu, H., and Xing, E. P. (2014). Graph estimation from multi-attribute data. *The Journal of Machine Learning Research* 15, 1713-1750.

Chen, J. and Chen, Z. (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika* 95, 759-771.

Examples

```
library(mvtnorm)
id <- rep(1:3, each=2)
Omega <- matrix(0, nrow=6, ncol=6)
Omega[1:4,1:4] <- -1
diag(Omega) <- 6
Sigma <- solve(Omega)
n <- 1000
Y <- rmvnorm(n, sigma=Sigma)
S <- crossprod(Y)

lambdas <- getLambdaRange(S, id, 3)
result <- multiAttSelect(S, n, id, lambdas)
```

perturbNodeTests

Get node-wise test statistics

Description

Given a vector id indicating node membership, test for perturbations at each node as described by Griffin et al. This function performs either the single-target test (sequential=FALSE, default) or the multi-target adjusted procedure (sequential=TRUE)

Usage

```
perturbNodeTests(Y, Omega, Sigma, id, sequential = FALSE,
  return.value = "stat")
```

Arguments

Y	matrix or data frame (rows=subjects, columns=variables)
Omega	precision matrix
Sigma	covariance matrix
id	vector mapping variables to nodes
sequential	boolean, whether or not to perform sequential adjustments
return.value	return statistics (stat) or p-values (pval)

Value

test statistics or p-values for perturbations at each individual node

References

Griffin, P.J., Johnson, W.E., and Kolaczyk, E.K. Detection of multiple perturbations in multi-omics biological networks (under review)

Examples

```
library(mvtnorm)
Omega <- matrix(0, nrow=6, ncol=6)
Omega[1:4,1:4] <- -1
diag(Omega) <- 6
Sigma <- solve(Omega)
mu <- c(1,1,0,0,0,0)
Y <- rmvnorm(20, mean=Sigma %*% mu, sigma=Sigma)
id <- c(1,1,2,2,3,3)
perturbNodeTests(Y, Omega, Sigma, id, sequential=FALSE) # unadjusted
perturbNodeTests(Y, Omega, Sigma, id, sequential=TRUE) # sequential
```

perturbTests

Test for perturbations at a series of locations

Description

Given a matrix of possible perturbation locations, perform the single-target (non-sequential) testing procedure described by Griffin et al. Return either test statistics or p-values.

Usage

```
perturbTests(Y, Omega, Sigma, perturb.mat, return.value = "stat")
```

Arguments

Y	matrix or data frame (rows=subjects, columns=variables)
Omega	precision matrix
Sigma	covariance matrix
perturb.mat	matrix indicating sets of which variables to test for perturbations
return.value	value to return (stat or pval)

Value

test statistics for perturbations at each of perturb.mat columns

References

Griffin, P.J., Johnson, W.E., and Kolaczyk, E.K. Detection of multiple perturbations in multi-omics biological networks (under review)

Examples

```
library(mvtnorm)
Omega <- matrix(0, nrow=6, ncol=6)
Omega[1:4,1:4] <- -1
diag(Omega) <- 6
Sigma <- solve(Omega)
mu <- c(1,1,0,0,0,0)
Y <- rmvnorm(20, mean=Sigma %*% mu, sigma=Sigma)
perturb.mat <- matrix(c(TRUE, TRUE, FALSE, FALSE, FALSE, FALSE,
                        FALSE, FALSE, TRUE, TRUE, FALSE, FALSE), ncol=2)
perturbTests(Y, Omega, Sigma, perturb.mat, return.value=stat)
```

seqPerturbTests

*Sequentially test for perturbations at a series of locations***Description**

Given a matrix of possible perturbation locations, perform the sequential testing procedure described by Griffin et al. Return either test statistics or p-values.

Usage

```
seqPerturbTests(Y, Omega, Sigma, perturb.mat, return.value = "stat")
```

Arguments

Y	matrix or data frame (rows=subjects, columns=variables)
Omega	precision matrix
Sigma	covariance matrix
perturb.mat	matrix indicating sets of which variables to test for perturbations
return.value	return statistics (stat) or p-values (pval)

Value

test statistics for perturbations at each of perturb.mat columns

References

Griffin, P.J., Johnson, W.E., and Kolaczyk, E.K. Detection of multiple perturbations in multi-omics biological networks (under review)

Examples

```
library(mvtnorm)
Omega <- matrix(0, nrow=6, ncol=6)
Omega[1:4,1:4] <- -1
diag(Omega) <- 6
Sigma <- solve(Omega)
mu <- c(1,1,0,0,0,0)
Y <- rmvnorm(20, mean=Sigma %*% mu, sigma=Sigma)
perturb.mat <- matrix(c(TRUE, TRUE, FALSE, FALSE, FALSE, FALSE,
                        FALSE, FALSE, TRUE, TRUE, FALSE, FALSE), ncol=2)
```

```
perturbTests(Y, Omega, Sigma, perturb.mat, return.value=stat) # initial ranking  
seqPerturbTests(Y, Omega, Sigma, perturb.mat, return.value=stat) # sequential adjusted
```


Index

`getLambdaRange`, [1](#)
`multiAttEstimate`, [2](#)
`multiAttLasso`, [3](#)
`multiAttSelect`, [4](#)
`perturbNodeTests`, [5](#)
`perturbTests`, [6](#)
`seqPerturbTests`, [7](#)