Package ‘sparseSEM’

February 20, 2015

Type Package
Title Sparse-aware Maximum Likelihood for Structural Equation Models
Version 2.5
Date 2013-09-03
Author Anhui Huang
Maintainer Anhui Huang <a.huang1@umiami.edu>
Description Sparse-aware maximum likelihood for structural equation models in inferring gene regulatory networks
License GPL
NeedsCompilation yes
Repository CRAN
Date/Publication 2014-09-04 07:49:26

R topics documented:

sparseSEM-package ........................................... 2
B ................................................................. 2
elasticNetSML ............................................... 3
elasticNetSMLcv ........................................... 5
elasticNetSMLpoint ......................................... 6
lassoSML ....................................................... 8
Missing ......................................................... 9
X ............................................................... 10
Y ............................................................... 10
Index ......................................................... 12
sparsesem-package

sparsesem: Sparse-aware Maximum Likelihood for Structural Equation Models in Inferring Gene Regulatory Networks

Description

state-of-the-art sparse aware maximum likelihood function for structural equation models. Two penalty functions including the Lasso and Elastic net available. Version 2.5: Fix a bug in internal function cv_gene_nets_support_adaENcv; no memory issue with Valgrind check.

Details

Package: sparseSEM
Type: Package
Version: 2.5
Date: 2014-09-03
License: GPL

Author(s)

Anhui Huang
Maintainer: Anhui Huang <a.huang1@umiami.edu>

References


Examples

library(sparseSEM)

B

True network edges

Description

B is the M by M matrix defining network topology
Usage
data(b)

Format
The format is: M by M, where M is the number of vertices (genes) num [1:30, 1:30] 0 0 0 0 0 ...

Details
If B is not available (real data): the stat output that describes the true accuracy and FDR should be ignored.

References

Examples
data(b)

data(8)

elasticNetSML  The Elastic Net penalty for SEM

Description
For each alpha from 0.95 to 0.05 at a step of 0.05, the function perform 5 fold CV for lambda_max to lambda_min in 20 step to determine the optimal alpha and lambda for the data.

Usage
elasticNetSML(Y, X, Missing, B, Verbose = 0)

Arguments
Y  gene expression M by N matrix
X  cis_eQTL M by N matrix
Missing  missing data in Y
B  true network topology if available
Verbose  describe the information output from 0 - 10, larger number means more output
Details

the function perform CV and parameter inference, calculate power and FDR

Value

Bout the matrix B from sparseSEM
fout f: the weight for matrix X
stat compute the power and FDR statistics if the true topology is provided
simTime computational time

Note

Difference in three functions:
1) elasticNetSML: Default alpha = 0.95: -0.05: 0.05; default 20 lambdas
2) elasticNetSMLcv: user supplied alphas (one or more), lambdas; compute the optimal parameters and network parameters
3) elasticNetSML-point: user supplied one alpha and one lambda, compute the network parameters

Author(s)

Anhui Huang; Dept of Electrical and Computer Engineering, Univ of Miami, Coral Gables, FL

References


Examples

library(sparsesem)
data(b);
data(y);
data(x);
data(missing);
Cexample is commented to satisfy CRAN's CPU time limit requirement.
#OUT <- elasticNetSML(y, x, Missing, B, Verbose = 1);
elasticNetSMLcv

_The Elastic Net penalty for SEM with user supplied alphas and lambdas_

**Description**

While elasticNetSML function has a set of default (alpha, lambda) and the optimal one is chosen by 5 fold cv, elasticNetSMLcv tests the combination of a set of alpha an lambda, and choose one as the optimal parameters. elasticNetSMLcv should be combined with elasticNetSMLpoint to obtain the network inference. For each alpha from the set of alphas provided, the function perform 5 fold CV for each user supplied lambda to determine the optimal alpha and lambda for the data.

**Usage**

`elasticNetSMLcv(Y, X, Missing, B, alpha_factors,lambda_factors, Verbose)`

**Arguments**

- `Y`: gene expression M by N matrix
- `X`: cis_eQTL M by N matrix
- `Missing`: missing data in Y
- `B`: true network topology if available
- `alpha_factors`: alpha_factors: the set of alphas to be tested, and is in range of (0, 1);
- `lambda_factors`: penalty lambda_factor: the set of lambda to be tested, and is in range of (0, 1);
- `Verbose`: describe the information output from 0 - 10, larger number means more output

**Details**

the function perform CV and parameter inference, calculate power and FDR

**Value**

- `Bout`: the matrix B from sparseSEM
- `fout`: f: the weight for matrix X
- `stat`: compute the power and FDR statistics if the ture topology is provided
- `simTime`: computational time
- `residual`: only meaningful for 1 alpha:
  - col1: lambdas;
  - col2: mean of residual error in k-fold CV
  - col3: standard error of residual error in k-fold CV
Note

Difference in three functions:
1) elasticNetSML: Default alpha = 0.95: -0.05: 0.05; default 20 lambdas
2) elasticNetSMLcv: user supplied alphas (one or more), lambdas; compute the optimal parameters and network parameters
3) elasticNetSMLpoint: user supplied one alpha and one lambda, compute the network parameters

Author(s)

Anhui Huang; Dept of Electrical and Computer Engineering, Univ of Miami, Coral Gables, FL

References


Examples

library(sparsesem)
data(B);
data(Y);
data(X);
data(Missing);
OUT <- elasticNetSMLcv(Y, X, Missing, B, alpha_factors = c(0.75, 0.5, 0.25), lambda_factors=c(0.1, 0.01, 0.001), Verbose = 1);

elasticNetSMLpoint(Y, X, Missing, B, alpha_factor, lambda_factor, Verbose)

Description

This function can be used after elasticNetSMLcv determines the optimal parameters. For user supplied one alpha in range of (0,1) and one lambda, the function perform selection path from lambda_max to lambda to determine the optimal network topology.

Usage

elasticNetSMLpoint(Y, X, Missing, B, alpha_factor, lambda_factor, Verbose)
Arguments

Y  gene expression M by N matrix
X  cis_eQTL M by N matrix
Missing  missing data in Y
B  true network topology if available
alpha_factor  alpha_factor: in range of (0, 1); must be scalar
lambda_factor  penalty lambda_factor: in range of (0, 1); must be scalar
Verbose  describe the information output from 0 - 10, larger number means more output

Details

the function perform selection path from lambda_max to lambda, calculate power and FDR

Value

Bout  the matrix B from sparseSEM
fout  f: the weight for matrix X
stat  compute the power and FDR statistics if the true topology is provided
simTime  computational time

Note

Difference in three functions:
1) elasticNetSML: Default alpha = 0.95: -0.05: 0.05; default 20 lambdas
2) elasticNetSMLcv: user supplied alphas (one or more), lambdas; compute the optimal parameters and network parameters
3) elasticNetSMLpoint: user supplied one alpha and one lambda, compute the network parameters

Author(s)

Anhui Huang; Dept of Electrical and Computer Engineering, Univ of Miami, Coral Gables, FL

References

Examples

```r
library(sparseSEM)
data(B);
data(Y);
data(X);
data(Missing);
OUT <- elasticNetSMLpoint(Y, X, Missing, B,
alpha_factor = 0.5, lambda_factor = 0.1, Verbose = 1);
```

---

**lassoSML**  
*The Lasso penalty for SML*

---

**Description**

Upon lambda_max to lambda_min in 20 step, the function compute 5 fold CV to determine the optimal lambda for the data.

**Usage**

```r
lassoSML(Y, X, Missing, B, Verbose = 5)
```

**Arguments**

- **Y**: gene expression M by N matrix  
- **X**: cis_eQTL M by N matrix  
- **Missing**: missing data in Y  
- **B**: true network topology if available  
- **Verbose**: describe the information output from 0 - 10, larger number means more output

**Details**

the function perform CV and parameter inference, calculate power and FDR

**Value**

- **Bout**: the matrix B from SEM  
- **fout**: f: the weight for matrix X  
- **stat**: compute the power and FDR statistics if the true topology is provided  
- **simTime**: computational time

**Author(s)**

Anhui Huang; Dept of Electrical and Computer Engineering, Univ of Miami, Coral Gables, FL
References


Examples

```r
library(sparseSEM)
data(B);
data(Y);
data(X);
data(Missing);
OUT <- lassoSML(Y, X, Missing, B, Verbose = 1);
```

Missing

*Missing gene expression data*

Description

M by N matrix corresponding to elements of Y. 0 denotes no missing, while 1 denotes missing

Usage

data(Missing)

Format

The format is: num [1:30, 1:200] 0 0 0 0 0 0 0 0 0 0 ... 

References


Examples

data(Missing)
**X**

*Genotype matrix*

**Description**

X is the M by N matrix corresponding to M cis-eQTL for M genes

**Usage**

```
data(X)
```

**Format**

The format is: int [1:30, 1:200] 2 1 3 1 2 3 2 1 2 2 ...

**Details**

current implementation only consider 1 eQTL per gene

**References**


**Examples**

```
data(X)
```

---

**Y**

*Gene expression matrix*

**Description**

Y is the M by N matrix describes the expression profile of M genes in N samples

**Usage**

```
data(Y)
```

**Format**

The format is: num [1:30, 1:200] 3.02 1.12 -2.24 3.58 2.18 ...

---
Details

Gene expression data

References


Examples

data(Y)
Index

*Topic **Elastic Net**
  elasticNetSML, 3
  elasticNetSMLcv, 5
  elasticNetSMLpoint, 6
*Topic **Lasso**
  lassoSML, 8
*Topic **datasets**
  B, 2
  Missing, 9
  X, 10
  Y, 10
*Topic **sparse aware maximum likelihood, structural equation models, Lasso, Elastic Net**
  sparseSEM-package, 2
*Topic **sparseSEM**
  elasticNetSML, 3
  elasticNetSMLcv, 5
  elasticNetSMLpoint, 6
  lassoSML, 8

B, 2

elasticNetSML, 3
elasticNetSMLcv, 5
elasticNetSMLpoint, 6

lassoSML, 8

Missing, 9

sparseSEM (sparseSEM-package), 2
sparseSEM-package, 2

X, 10

Y, 10