Package 'sparseSEM'

October 27, 2024

Type Package

Title Elastic Net Penalized Maximum Likelihood for Structural Equation Models with Network GPT Framework

Version 4.1

Date 2024-10-25

Maintainer Anhui Huang <anhuihuang@gmail.com>

Description Provides elastic net penalized maximum likelihood estimator for structural equation models (SEM). The package implements `lasso` and `elastic net` (l1/l2) penalized SEM and estimates the model parameters with an efficient block coordinate ascent algorithm that maximizes the penalized likelihood of the SEM. Hyperparameters are inferred from cross-validation (CV). A Stability Selection (STS) function is also available to provide accurate causal effect selection. The software achieves high accuracy performance through a `Network Generative Pre-trained Transformer` (Network GPT) Framework with two steps: 1) pretrains the model to generate a complete (fully connected) graph; and 2) uses the complete graph as the initial state to fit the `elastic net` penalized SEM.

Depends R (>= 3.5.0)

Imports parallel

License GPL

NeedsCompilation yes

Repository CRAN

Suggests knitr,plot.matrix

VignetteBuilder knitr

Date/Publication 2024-10-27 15:30:02 UTC

Author Anhui Huang [aut, ctb, cre]

Contents

`sparseSEM-package` *sparseSEM: Elastic Net Penalized Maximum Likelihood for Structural Equation Models with Network GPT Framework*

Description

State-of-the-art Elastic Net Penalized Maximum Likelihood for Structural Equation Models implemented with a Network Generative Pre-training Transformer (Network GPT). Two penalty functions including Lasso and Elastic-net are available.

For users new to this package, function elasticNetSEM() provides the simplified entry point: Missing matrix can be all 0 (none or uknown), so as B matrix (unknow connections in the network), thus only Y and X are mandatory. Then model will fit SEM: Y = BY + fX + e. See the reference for model details.

The package also provides other functions with more flexibility to allow fine tuning the parameters:
- elasticNetSEMcv(): user provides alphas (one or more) and lambdas; the function then computes the optimal parameters and network parameters;
- elasticNetSEMpoint(): user provides one pair of (alpha, lambda), and the function computes the network parameters.
- enSEM_stability_selection(): stability selection via bootstrapping.

Network Generative Pre-training Transformer (GPT) Framework:
In all functions the "Network GPT" framework is deployed behind the scene. Specifically, a pre-trained network was built in the following steps:

- Step 1. Pre-train the model with ridge (L2 penalty) SEM with k-fold CV: this step find the optimal ridge hyperparameter rho;

- Step 2. Generate a complete graph by fitting the SEM ridge regression model with rho from Step 1, obtain the initial status of a (non-sparse, fully connected) complete network structure (B_hat).

Note that the term "Transformer" does not carry the same meaning as the "transformer architecture" commonly used in Natural Language Processing (NLP). In Network GPT, the term means the creation and generation of the complete graph.

Regularization path: A lasso-strong rule is developed for SEM and applied to the selection path. In each step from lambda_max to lambda_min, where lambda_max is the lambda that keeps only 1 non-zero edge, and lambda_min is the smallest lambda set arbitrarily (eg., from CV, or 0.001 *

lambda_max), the elements in B are pre-set to 0 if they meet with the discarding rule. Those elements will not be computed again in the block coordinate ascent algorithm, resulting in reducing computational costs. See the Vignettes and references for more details.

Details

Package:	sparseSEM
Type:	Package
Version:	4.1
Date:	2024-10-25
License:	GPL

Author(s)

Anhui Huang

Maintainer: Anhui Huang <anhuihuang@gmail.com>

References

1. Cai, X., Bazerque, J.A., and Giannakis, G.B. (2013). Inference of Gene Regulatory Networks with Sparse Structural Equation Models Exploiting Genetic Perturbations. PLoS Comput Biol 9, e1003068.
2. Huang, A. (2014). "Sparse model learning for inferring genotype and phenotype associations." Ph.D Dissertation. University of Miami(1186).

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 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

1. Cai, X., Bazerque, J.A., and Giannakis, G.B. (2013). Inference of Gene Regulatory Networks with Sparse Structural Equation Models Exploiting Genetic Perturbations. PLoS Comput Biol 9, e1003068.
2. Huang, A. (2014). "Sparse model learning for inferring genotype and phenotype associations." Ph.D Dissertation. University of Miami(1186).

Examples

```
data(B)
```

`elasticNetSEM` *The Elastic Net penalized SEM with Network GPT Framework*

Description

Fit the elastic-net penalized structureal Equation Models (SEM) with input data (X, Y): Y = BY + fX + e.

For users new to this package, elasticNetSEM provides the simplified entry point: Missing matrix can be all 0 (none or uknown), so as B matrix (unknow connections in the network), thus only Y and X are mandatory.

Underlying the function, the program obtains the optimal hyperparameter (alpha, lambda) from k-fold cross validation (CV) with fixed k= 5. Specifically, 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, lambda) for the data.

Generally, the software program performs the following Network GPT Framework to arrive at final network structure:
Step 1. Generating a Complete Graph:

- SEM-ridge regression (L2 penalty) with k-fold CV: this step find the optimal ridge hyperparameter rho;

- fit SEM ridge regression model (L2 penalty) with rho from Step 1, obtain the initial status (non-sparse) of network structure (B_ridge);

Step 2. Elastic net penalized SEM regression with k-fold CV: this step finds the optimal hyperparameter (alpha, lambda);

Step 3. Fit elastic net SEM model with (alpha, lambda) from Step 2; This step applies a block cooridnate ascent algorithm, and the complete graph from Step-1 is used as the intial step;

Step 4. Calculate results for PD, FDR, provide the function output.

For large scale network inference, a standalone C/C++ software with openMPI for parallel computation is also available upon request.

Usage

```
elasticNetSEM(Y, X, Missing, B, verbose = 0)
```

Arguments

`Y`	The observed node response data with dimension of M (nodes) by N (samples). Y is normalized inside the function.
`X`	The network node attribute matrix with dimension of M by N. Theoretically, X can be L by N matrix, with L being the total node attributes. In current implementation, each node only allows one and only one attribute. If you have more than one attributes for some nodes, please consider selecting the top one by either correlation or principal component methods. If for some nodes there is no attribute available, fill in the rows with all zeros. See the yeast data 'yeast.rda' for example. X is normalized inside the function.
`Missing`	Optional M by N matrix corresponding to elements of Y. 0 denotes not missing, and 1 denotes missing. If a node i in sample j has the label missing (Missing[i,j] = 1), then Y[i,j] is set to 0.
`B`	Optional input. For a network with M nodes, B is the M by M adjacency matrix. If data is simulated/with known true network topology (i.e., known adjacency matrix), the Power of detection (PD) and False Discovery Rate (FDR) is computed in the output parameter 'statistics'. If the true network topology is unknown, B is optional, and the PD/FDR in output parameter 'statistics' should be ignored.
`verbose`	describe the information output from -1 - 10, larger number means more output

Details

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

Value

`Bout`	the computed weights for the network topology. B[i,j] = 0 means there is no edge between node i and j; B[i,j]!=0 denotes an (undirected) edge between note i and j with B[i,j] being the weight of the edge.

fout	f is 1 by M array keeping the weight for X (in SEM: Y = BY + FX + e). Theoretically, F can be M by L matrix, with M being the number of nodes, and L being the total node attributes. However, in current implementation, each node only allows one and only one attribute. If you have more than one attributes for some nodes, please consider selecting the top one by either correlation or principal component methods.
stat	statistics is 1x6 array keeping record of: 1. correct positive 2. total positive 3. false positive 4. positive detected 5. Power of detection (PD) = correct positive/total positive 6. False Discovery Rate (FDR) = false positive/positive detected
hyperparameters	Model hyperparameters obtained from cross validation.
runTime	computational time
call	the call that produced this object

Note

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

1. Cai, X., Bazerque, J.A., and Giannakis, G.B. (2013). Inference of Gene Regulatory Networks with Sparse Structural Equation Models Exploiting Genetic Perturbations. PLoS Comput Biol 9, e1003068.
2. Huang, A. (2014). "Sparse model learning for inferring genotype and phenotype associations." Ph.D Dissertation Chapter 7. University of Miami(1186).

Examples

```
library(sparseSEM)
data(B);
data(Y);
data(X);
data(Missing);
#Example

  OUT <- elasticNetSEM(Y, X, Missing, B, verbose = 1);
```

`elasticNetSEMcv` *The Elastic Net penalty for SEM with user supplied (alphas, lambdas) for grid search*

Description

Function elasticNetSEMcv allows users to set their own grid search through combination of a set of user provided alphas an lambdas.

Usage

```
elasticNetSEMcv(Y, X, Missing, B, alpha_factors,lambda_factors,kFold, verbose)
```

Arguments

`Y`	The observed node response data with dimension of M (nodes) by N (samples). Y is normalized inside the function.
`X`	The network node attribute matrix with dimension of M by N. Theoretically, X can be L by N matrix, with L being the total node attributes. In current implementation, each node only allows one and only one attribute. If you have more than one attributes for some nodes, please consider selecting the top one by either correlation or principal component methods. If for some nodes there is no attribute available, fill in the rows with all zeros. See the yeast data 'yeast.rda' for example. X is normalized inside the function.
`Missing`	Optional M by N matrix corresponding to elements of Y. 0 denotes not missing, and 1 denotes missing. If a node i in sample j has the label missing (Missing[i,j] = 1), then Y[i,j] is set to 0.
`B`	Optional input. For a network with M nodes, B is the M by M adjacency matrix. If data is simulated/with known true network topology (i.e., known adjacency matrix), the Power of detection (PD) and False Discovery Rate (FDR) is computed in the output parameter 'statistics'. If the true network topology is unknown, B is optional, and the PD/FDR in output parameter 'statistics' should be ignored.
`alpha_factors`	The set of candidate alpha values. Default is seq(start = 0.95, to = 0.05, step = -0.05)
`lambda_factors`	The set of candidate lambda values. Default is 10^seq(start =1, to = 0.001, step = -0.2)
`kFold`	k-fold cross validation, default k=5
`verbose`	describe the information output from -1 - 10, larger number means more output

Details

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

Value

cv dataframe stores the minimum Mean Square Error (MSE) for each alpha and the corresponding lambda from the selection path [lambda_max, ..., lambda_min].
col1: alpha
col2: lambda (With the given alpha, this is the lambda having minimum MSE)
col3: MSE
col4: STE

The final (alpha, lambda) is set at the (alpha, lambda) that is within 1ste of the min(MSE) with higher level of penalty on the likehood function.

- fit the model fit with optimal (alpha,lambda) from cv
  - Bout the computed weights for the network topology. B[i,j] = 0 means there is no edge between node i and j; B[i,j]!=0 denotes an (undirected) edge between note i and j with B[i,j] being the weight of the edge.
  - fout f is 1 by M array keeping the weight for X (in SEM: Y = BY + FX + e). Theoretically, F can be M by L matrix, with M being the number of nodes, and L being the total node attributes. However, in current implementation, each node only allows one and only one attribute. If you have more than one attributes for some nodes, please consider selecting the top one by either correlation or principal component methods.
  - stat statistics is 1x6 array keeping record of:
    1. correct positive
    2. total positive
    3. false positive
    4. positive detected
    5. Power of detection (PD) = correct positive/total positive
    6. False Discovery Rate (FDR) = false positive/positive detected
  - simTime computational time
  - call the call that produced this object

Note

Difference in three functions:
1) elasticNetSML: Default alpha = 0.95: -0.05: 0.05; default 20 lambdas
2) elasticNetSEMcv: 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

User is responsible to set the random seed to guarantee repeatable results.

Author(s)

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

References

1. Cai, X., Bazerque, J.A., and Giannakis, G.B. (2013). Inference of Gene Regulatory Networks with Sparse Structural Equation Models Exploiting Genetic Perturbations. PLoS Comput Biol 9, e1003068.
2. Huang, A. (2014). "Sparse model learning for inferring genotype and phenotype associations." Ph.D Dissertation. University of Miami(1186).

Examples

```
library(sparseSEM)
data(B);
data(Y);
data(X);
data(Missing);
## Not run: OUT <- elasticNetSEMcv(Y, X, Missing, B, alpha_factors = c(0.75, 0.5, 0.25),
lambda_factors=c(0.1, 0.01, 0.001), kFold = 5, verbose  = 1);

## End(Not run)
```

---

elasticNetSEMpoint *The Elastic Net penalty for SEM*

Description

For user provided one alpha in range (0,1) and one lambda_factor in range (0,1), the function perform selection path from lambda_max to lambda to determine the optimal network topology.

In the case of the grid search in elasticNetSEMcv() function may not be granular enough and user would like to explore/twist (alpha, lambda) a little bit, this function provides the solution.

Usage

```
elasticNetSEMpoint(Y, X, Missing, B, alpha_factor, lambda_factor, verbose)
```

Arguments

Y	The observed node response data with dimension of M (nodes) by N (samples). Y is normalized inside the function.
X	The network node attribute matrix with dimension of M by N. Theoretically, X can be L by N matrix, with L being the total node attributes. In current implementation, each node only allows one and only one attribute. If you have more than one attributes for some nodes, please consider selecting the top one by either correlation or principal component methods. If for some nodes there is no attribute available, fill in the rows with all zeros. See the yeast data 'yeast.rda' for example. X is normalized inside the function.

Missing	Optional M by N matrix corresponding to elements of Y. 0 denotes not missing, and 1 denotes missing. If a node i in sample j has the label missing (Missing[i,j] = 1), then Y[i,j] is set to 0.
B	Optional input. For a network with M nodes, B is the M by M adjacency matrix. If data is simulated/with known true network topology (i.e., known adjacency matrix), the Power of detection (PD) and False Discovery Rate (FDR) is computed in the output parameter 'statistics'. If the true network topology is unknown, B is optional, and the PD/FDR in output parameter 'statistics' should be ignored.
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 -1 - 10, larger number means more output

Details

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

Value

Bout	the computed weights for the network topology. B[i,j] = 0 means there is no edge between node i and j; B[i,j]!=0 denotes an (undirected) edge between note i and j.
fout	f is 1 by M array keeping the weight for X (in SEM: Y = BY + FX + e). Theoretically, F can be M by L matrix, with M being the number of nodes, and L being the total node attributes. However, in current implementation, each node only allows one and only one attribute. If you have more than one attributes for some nodes, please consider selecting the top one by either correlation or principal component methods.
stat	statistics is 1x6 array keeping record of: 1. correct positive 2. total positive 3. false positive 4. positive detected 5. Power of detection (PD) = correct positive/total positive 6. False Discovery Rate (FDR) = false positive/positive detected
simTime	computational time
call	the call that produced this object

Note

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

1. Cai, X., Bazerque, J.A., and Giannakis, G.B. (2013). Inference of Gene Regulatory Networks with Sparse Structural Equation Models Exploiting Genetic Perturbations. PLoS Comput Biol 9, e1003068.
2. Huang, A. (2014). "Sparse model learning for inferring genotype and phenotype associations." Ph.D Dissertation. University of Miami(1186).

Examples

```
library(sparseSEM)
data(B);
data(Y);
data(X);
data(Missing);
## Not run:  OUT <- elasticNetSEMpoint(Y, X, Missing, B,
alpha_factor = 0.5, lambda_factor = 0.1, verbose = 1);

## End(Not run)
```

---

enSEM_stability_selection

Stability Selection for the Elastic Net penalized SEM

Description

Fit the elastic-net penalized structureal Equation Models (SEM) with input data (X, Y): Y = BY + fX + e. Perform Stability Selection (STS) on the input dataset. This function implements STS described in Meinshausen N. and Buhlmann P (2010) and Shah R. and Samworth R (2013).

Underlying the function, the program obtains the performs n rounds of boostraping each with half of the original sample size, and run the selection path of hyperparameter (alpha, lambda). The following stability selection scores are calculated:
1. E(v): the upper bound of the expected number of falsely selected variables
2. pre-comparison error rate = E(v)/p where p is the total number of model parameters (in SEM, p = M*M -M)
3. E(v)_ShaR the expected number of falsely selected variables described in Shah R. and Samworth R (2013)
4. FDR: False discovery rate = E(v)/nSelected
5. FDR_ShaR: FDR described in Shah R. and Samworth R (2013)

The final output is based on Scores described in described in Shah R. and Samworth R (2013), and original scores described in Meinshausen N. and Buhlmann P (2010) are provided for reference.

Usage

```
enSEM_stability_selection(Y,X, Missing,B,
                          alpha_factors,
                          lambda_factors,
                          kFold,
                          nBootstrap,
                          verbose)
```

Arguments

Y	The observed node response data with dimension of M (nodes) by N (samples). Y is normalized inside the function.
X	The network node attribute matrix with dimension of M by N. Theoretically, X can be L by N matrix, with L being the total node attributes. In current implementation, each node only allows one and only one attribute. If you have more than one attributes for some nodes, please consider selecting the top one by either correlation or principal component methods. If for some nodes there is no attribute available, fill in the rows with all zeros. See the yeast data 'yeast.rda' for example. X is normalized inside the function.
Missing	Optional M by N matrix corresponding to elements of Y. 0 denotes not missing, and 1 denotes missing. If a node i in sample j has the label missing (Missing[i,j] = 1), then Y[i,j] is set to 0.
B	Optional input. For a network with M nodes, B is the M by M adjacency matrix. If data is simulated/with known true network topology (i.e., known adjacency matrix), the Power of detection (PD) and False Discovery Rate (FDR) is computed in the output parameter 'statistics'. If the true network topology is unknown, B is optional, and the PD/FDR in output parameter 'statistics' should be ignored.
alpha_factors	The set of candidate alpha values. Default is seq(start = 0.95, to = 0.05, step = -0.05)
lambda_factors	The set of candidate lambda values. Default is 10^seq(start =1, to = 0.001, step = -0.2)
kFold	k-fold cross validation, default k=3. Note STS result is not based on CV. However, fitting l1/l2 regularized SEM will run the first step described in elasticNetSEM() function: Step 1. SEM-ridge regression (L2 penalty) with k-fold CV: this step find the optimal ridge hyperparameter rho to provide an initial values for l1/l2 regularized SEM.
nBootstrap	bootstrapping parameter. default nBootstrap = 100.
verbose	describe the information output from -1 - 10, larger number means more output

Details

the function perform STS

Value

STS	The stable effects are those effects selected by STS, i.e., the non-zero values in matrix B.
statistics	the final STS scores with components of: 1. threshold: denoted as pi in Meinshausen N. and Buhlmann P (2010) 2. pre-comparison error rate 3. E(v) 4. E(v)_ShahR 5. nSTS: final number of stable effects with pi that leads to minimum FDR 6. FDR 7. FDR_ShahR
STS data	Bootstrapping details.
call	the call that produced this object

Author(s)

Anhui Huang

References

[1]: Meinshausen, N. and Buhlmann, P., 2010. Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(4), pp.417-473.

[2] Shah, R.D. and Samworth, R.J., 2013. Variable selection with error control: another look at stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 75(1), pp.55-80.

Examples

```
library(sparseSEM)
data(B);
data(Y);
data(X);
data(Missing);
#Example

output = enSEM_stability_selection(Y,X, Missing,B,
                                   alpha_factors = seq(1,0.05, -0.05),
                                   lambda_factors =10^seq(-0.2,-4,-0.2),
                                   kFold = 3,
                                   nBootstrap = 100,
                                   verbose = -1)
```

`enSEM_stability_selection_parallel` *Parallel Stability Selection for the Elastic Net penalized SEM*

Description

Fit the elastic-net penalized structureal Equation Models (SEM) with input data (X, Y): Y = BY + fX + e. Perform Stability Selection (STS) on the input dataset. This function implements STS described in Meinshausen N. and Buhlmann P (2010) and Shah R. and Samworth R (2013).

Underlying the function, the program obtains the performs n rounds of boostraping each with half of the original sample size, and run the selection path of hyperparameter (alpha, lambda). The following stability selection scores are calculated:
1. E(v): the upper bound of the expected number of falsely selected variables
2. pre-comparison error rate = E(v)/p where p is the total number of model parameters (in SEM, p = M*M -M)
3. E(v)_ShaR the expected number of falsely selected variables described in Shah R. and Samworth R (2013)
4. FDR: False discovery rate = E(v)/nSelected
5. FDR_ShaR: FDR described in Shah R. and Samworth R (2013)

The final output is based on Scores described in described in Shah R. and Samworth R (2013), and original scores described in Meinshausen N. and Buhlmann P (2010) are provided for reference.

This function 'enSEM_stability_selection_parallel' performs the same computation as that in function 'enSEM_stability_selection' with the only difference of setting up the bootstrapping in parallel leveraging the 'parallel' package.

Usage

```
enSEM_stability_selection_parallel(Y,X, Missing,B,
                                   alpha_factors,
                                   lambda_factors,
                                   kFold,
                                   nBootstrap,
                                   verbose,
                                   clusters)
```

Arguments

`Y`	The observed node response data with dimension of M (nodes) by N (samples). Y is normalized inside the function.
`X`	The network node attribute matrix with dimension of M by N. Theoretically, X can be L by N matrix, with L being the total node attributes. In current implementation, each node only allows one and only one attribute. If you have more than one attributes for some nodes, please consider selecting the top one by either correlation or principal component methods. If for some nodes there is no attribute available, fill in the rows with all zeros.

	See the yeast data 'yeast.rda' for example. X is normalized inside the function.
Missing	Optional M by N matrix corresponding to elements of Y. 0 denotes not missing, and 1 denotes missing. If a node i in sample j has the label missing (Missing[i,j] = 1), then Y[i,j] is set to 0.
B	Optional input. For a network with M nodes, B is the M by M adjacency matrix. If data is simulated/with known true network topology (i.e., known adjacency matrix), the Power of detection (PD) and False Discovery Rate (FDR) is computed in the output parameter 'statistics'. If the true network topology is unknown, B is optional, and the PD/FDR in output parameter 'statistics' should be ignored.
alpha_factors	The set of candidate alpha values. Default is seq(start = 0.95, to = 0.05, step = -0.05)
lambda_factors	The set of candidate lambda values. Default is 10^seq(start =1, to = 0.001, step = -0.2)
kFold	k-fold cross validation, default k=3. Note STS result is not based on CV. However, fitting l1/l2 regularized SEM will run the first step described in elasticNetSEM() function: Step 1. SEM-ridge regression (L2 penalty) with k-fold CV: this step find the optimal ridge hyperparameter rho to provide an initial values for l1/l2 regularized SEM.
nBootstrap	bootstrapping parameter. default nBootstrap = 100.
verbose	describe the information output from -1 - 10, larger number means more output
clusters	snow clusters

Details

the function perform STS

Value

STS	The stable effects are those effects selected by STS, i.e., the non-zero values in matrix B.
statistics	the final STS scores with components of: 1. threshold: denoted as pi in Meinshausen N. and Buhlmann P (2010) 2. pre-comparison error rate 3. E(v) 4. E(v)_ShahR 5. nSTS: final number of stable effects with pi that leads to minimum FDR 6. FDR 7. FDR_ShahR
STS data	Bootstrapping details.
call	the call that produced this object

Author(s)

Anhui Huang

References

[1]: Meinshausen, N. and Buhlmann, P., 2010. Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(4), pp.417-473.

[2] Shah, R.D. and Samworth, R.J., 2013. Variable selection with error control: another look at stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 75(1), pp.55-80.

Examples

```
library(sparseSEM)
library(parallel)
data(B);
data(Y);
data(X);
data(Missing);
#Example


  cl<-makeCluster(2)
  clusterEvalQ(cl,{library(sparseSEM)})
  output = enSEM_stability_selection_parallel(Y,X, Missing,B,
                                              alpha_factors = seq(1,0.05, -0.05),
                                              lambda_factors =10^seq(-0.2,-4,-0.2),
                                              kFold = 3,
                                              nBootstrap = 100,
                                              verbose = -1,
                                              clusters = cl)
  stopCluster(cl)
```

`lassoSEM` *The Lasso penalty for SEM*

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

```
lassoSEM(Y, X, Missing, B, verbose = 5)
```

Arguments

Y	gene expression M by N matrix
X	The network node attribute matrix with dimension of M by N. Theoretically, X can be L by N matrix, with L being the total node attributes. In current implementation, each node only allows one and only one attribute. If you have more than one attributes for some nodes, please consider selecting the top one by either correlation or principal component methods. If for some nodes there is no attribute available, fill in the rows with all zeros. See the yeast data 'yeast.rda' for example. X is normalized inside the function.
Missing	missing data in Y
B	true network topology if available
verbose	describe the information output from -1 - 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 ture topology is provided
simTime	computational time

Author(s)

Anhui Huang

References

1. Cai, X., Bazerque, J.A., and Giannakis, G.B. (2013). Inference of Gene Regulatory Networks with Sparse Structural Equation Models Exploiting Genetic Perturbations. PLoS Comput Biol 9, e1003068.
2. Huang, A. (2014). "Sparse model learning for inferring genotype and phenotype associations." Ph.D Dissertation. University of Miami(1186).

Examples

```
library(sparseSEM)
data(B);
data(Y);
data(X);
data(Missing);
## Not run: OUT <- lassoSEM(Y, X, Missing, B, verbose = 0);
```

Missing *Missing Network Node dependent variable 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

1. Cai, X., Bazerque, J.A., and Giannakis, G.B. (2013). Inference of Gene Regulatory Networks with Sparse Structural Equation Models Exploiting Genetic Perturbations. PLoS Comput Biol 9, e1003068.
2. Huang, A. (2014). "Sparse model learning for inferring genotype and phenotype associations." Ph.D Dissertation. University of Miami(1186).

Examples

```
data(Missing)
```

X *Genotype matrix*

Description

X is the M by N matrix corresponding to M network nodes from N samples.

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 inpepedent attribute per node. If users have more than one attributes for some nodes, please consider selecting the top one by either correlation or principal component methods.

References

1. Cai, X., Bazerque, J.A., and Giannakis, G.B. (2013). Inference of Gene Regulatory Networks with Sparse Structural Equation Models Exploiting Genetic Perturbations. PLoS Comput Biol 9, e1003068.
2. Huang, A. (2014). "Sparse model learning for inferring genotype and phenotype associations." Ph.D Dissertation. University of Miami(1186).

Examples

```
data(X)
```

Y *Gene expression matrix*

Description

Y is the M by N matrix describes the dependent attribute of M nodes 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

1. Cai, X., Bazerque, J.A., and Giannakis, G.B. (2013). Inference of Gene Regulatory Networks with Sparse Structural Equation Models Exploiting Genetic Perturbations. PLoS Comput Biol 9, e1003068.
2. Huang, A. (2014). "Sparse model learning for inferring genotype and phenotype associations." Ph.D Dissertation. University of Miami(1186).

Examples

```
data(Y)
```

`yeast` *Yeast cis-QTL Gene Regulatory Network Dataset*

Description

The dataset (Y,X) are two matrices each with 3380 rows and 112 columns as described in the Vignettee "Elastic Net Enabled Sparse-Aware Maximum Likelihood for Structural Equation Models in Inferring Gene Regulatory Networks".

The Yeast expression trait data set was obatained from Brem R.B. and Kruglyak L (2005), and has been screened through:
1. screen and keep the ORF names in Kellis's ORF list (Kellis M et al, 2003);
2. screen out ORF with more than 5% of missing expression date;
3. perform eQTL mapping by Wilcoxon test, adjust by qvalue; keep the top one cisQTL;
4. fill in zeros for ORF without cisQTL in matrix X

Usage

```
data(yeast)
```

Format

The format is matrix.

Details

Yeast cis-QTL Gene Regulatory Network Dataset

References

1. Brem RB, Kruglyak L: The landscape of genetic complexity across 5,700 gene expression traits in yeast. Proceedings of the National Academy of Sciences of the United States of America 2005, 102:1572-1577.
2. Kellis M, Patterson N, Endrizzi M, Birren B, Lander ES: Sequencing and comparison of yeast species to identify genes and regulatory elements. Nature 2003, 423:241-254

Examples

```
data(yeast)
```

Index