Package ‘hmgm’

December 16, 2019

Type Package

Title High-Dimensional Mixed Graphical Models Estimation

Version 1.0.2

Author Mingyu Qi, Tianxi Li

Maintainer Mingyu Qi <mq3sq@virginia.edu>

Description
Provides weighted group lasso framework for high-dimensional mixed data graph estimation.
In the graph estimation stage, the graph structure is estimated by maximizing the conditional
likelihood of one variable given the rest. We focus on the conditional loglikelihood of each variable
and fit separate regressions to estimate the parameters, much in the spirit of the neighborhood
selection approach proposed by Meinshausen-
Buhlmann for the Gaussian Graphical Model and by Ravikumar
for the Ising Model. Currently, the discrete variables can only take two values. In the future, method
for general discrete data and for visualizing the estimated graph will be added.
For more details, see the linked paper.

URL <https://arxiv.org/pdf/1304.2810.pdf>

License GPL (>= 2)

Depends R(>= 3.5.0)

Imports rgl, Matrix, glmnet, MASS, nat, binaryLogic, Rcpp, stats,
methods

NeedsCompilation yes

Encoding UTF-8

RoxygenNote 7.0.1

Repository CRAN

Date/Publication 2019-12-16 14:30:15 UTC

R topics documented:

`hmgm-package` *High-dimensional mixed graphical models estimation*

Description

A package for high-dimensional mixed graphical models estimation.

Details

Package:	hmgm
Type:	Package
Version:	0.3.0
Date:	2019-11-29
License:	GPL (>= 2)

The major function `hmgm` provides weighted group lasso framework for high-dimensional mixed data graph estimation.

Another function `pargroup` identify all regions where groups intersect, make all variables in each overlapping region into a new group.

Author(s)

Mingyu Qi, Tianxi Li

References

Jie Cheng, Tianxi Li, Elizaveta Levina, and Ji Zhu.(2017) *High-dimensional Mixed Graphical Models. Journal of Computational and Graphical Statistics 26.2 (2017): 367-378*,https://arxiv.org/pdf/1304.2810.pdf
Simon, N., Friedman, J., Hastie,T., Tibshirani, R.(2011) *Regularization Paths for Cox's ProportionalHazards Model via Coordinate Descent, Journal of Statistical Software, Vol.39(5) 1-13*,https://www.jstatsoft.org/v39/i05/
Meinshausen, N. and Buhlmann, P. (2006) *High dimensional graphs and variable selection with the lasso, Annals of Statistics, 34, 1436–1462.*, https://arxiv.org/pdf/math/0608017.pdf
Ravikumar, P., Wainwright, M., and Lafferty, J.(2010) *High-dimensionalIsing model selection using l1-regularized logistic regression,Annals of Statistics, 38, 1287–1319.*,https://arxiv.org/pdf/1010.0311.pdf

Liu, H., Han, F., Yuan, M., Lafferty, J., and Wasserman, L.(2012) *High dimensional semiparametric Gaussian copula graphical models, Annals of Statistics, 40, 2293–2326.*, https://arxiv.org/pdf/1202.2169.pdf

Zhao, P., Rocha, G., and Yu, B.(2009) *The composite absolute penalties family for grouped and hierarchical variable selection, The Annals of Statistics, 3468–3497.*, https://arxiv.org/pdf/0909.0411.pdf

---

`datagen` *Data generator*

Description

The data generator creates random samples from conditional Gaussian distribution with different graph structures

Usage

```
datagen(parlist,n)
```

Arguments

`parlist`	The parameter list generated by `pargen`
`n`	The number of observations (sample size)

Details

We use the exact probability rather than `MCMC` methods to generate the binary variables. We generate the probability distribution of `Z` as well as the canonical parameters. The memory requirements for the distribution of `Z` make it difficult to generate a large number of binary variables in simulations. However, this is not a problem for real data where the variables are already observed.

Value

The function returns a data list:

`z`	Value of binary variable
`y`	Value of continous variable
`Prob`	The probability distribution of discrete variables
`cparlist`	The canonical parameter

Author(s)

Mingyu Qi, Tianxi Li

References

Jie Cheng, Tianxi Li, Elizaveta Levina, and Ji Zhu.(2017) *High-dimensional Mixed Graphical Models. Journal of Computational and Graphical Statistics 26.2: 367-378*, https://arxiv.org/pdf/1304.2810.pdf

See Also

pargen

Examples

```
#set parameters
n = 100
p = 20
q = 10
a = 1
b = 2
c = 1
adj = matrix(0, p+q, p+q)
adj[10:16, 10:16] = 1
adj[1:5, 1:5] = 1
adj[25:30, 25:30] = 1
adj = adj-diag(diag(adj))
parlist = pargen(adj, p, q, a, b, c)

#generate data
mydata = datagen(parlist, n)
```

---

edgenorm *Calculate the group L2 norm for each pair of edges*

Description

Function to calculate the group L2 norm for each pair of edges

Usage

```
edgenorm(fitlistpost)
```

Arguments

fitlistpost	The fitted parameter path

Value

The function returns a list of group L2 norm for each pair of edges

zz	Group L2 norm for each pair of edges connecting binary variables
zy	Group L2 norm for each pair of edges connecting binary variables and continous variables
yy	Group L2 norm for each pair of edges connecting continous variables

Author(s)

Mingyu Qi, Tianxi Li

References

Jie Cheng, Tianxi Li, Elizaveta Levina, and Ji Zhu. (2017) *High-dimensional Mixed Graphical Models. Journal of Computational and Graphical Statistics 26.2: 367-378*, https://arxiv.org/pdf/1304.2810.pdf

See Also

hmgm

Examples

```
n = 100
p = 20
q = 10
a = 1
b=  2
c = 1


adj = matrix(0, p+q, p+q)
adj[10:16, 10:16] = 1
adj[1:5, 1:5] = 1
adj[25:30, 25:30] = 1
adj = adj-diag(diag(adj))

parlist = pargen(adj, p, q, a, b,c)

mydata = datagen(parlist, n)

z = mydata$z

y = mydata$y

tune1 = tune2 = 0.1

kappa = 0.1

## parameter estimation
```

```
fit = hmgm(z, y, tune1, tune2, 'max', kappa)

##calculate the group L2 norm for each pair of edges

fitlist_post = fit$fitlist_post
adj_norm = edgenorm(fitlist_post)
```

---

fitadj *Obtain the adjascent matrix by thresholding the adj norm matrix*

Description

Function to obtain the adjascent matrix by thresholding the adj norm matrix

Usage

```
fitadj(adj_norm, thres)
```

Arguments

adj_norm	A structure with adj norm matrix zz zy yy
thres	Length of thresholding vector

Value

The function returns a 4-dimentional array to record the adj matrix.

Author(s)

Mingyu Qi, Tianxi Li

References

Jie Cheng, Tianxi Li, Elizaveta Levina, and Ji Zhu. (2017) *High-dimensional Mixed Graphical Models. Journal of Computational and Graphical Statistics 26.2: 367-378*, https://arxiv.org/pdf/1304.2810.pdf

See Also

hmgm edgenorm

Examples

```
n = 100
p = 20
q = 10
a = 1
b=  2
c = 1


adj = matrix(0, p+q, p+q)
adj[10:16, 10:16] = 1
adj[1:5, 1:5] = 1
adj[25:30, 25:30] = 1
adj = adj-diag(diag(adj))

parlist = pargen(adj, p, q, a, b,c)

mydata = datagen(parlist, n)

z = mydata$z

y = mydata$y

tune1 = tune2 = 0.1

kappa = 0.1

## parameter estimation

fit = hmgm(z, y, tune1, tune2, 'max',kappa)

#calculate the group L2 norm for each pair of edges

fitlist_post = fit$fitlist_post
adj_norm = edgenorm(fitlist_post)

adj_lambda = fitadj(adj_norm, 0)
```

---

hmgm *High-dimensional Mixed Graphical Models Estimation*

Description

The main function for high-dimensional Mixed Graphical Models estimation.

Usage

```
hmgm(z,y,tune1,tune2,method,kappa,penalty1=NULL,penalty2=NULL)
```

Arguments

`z`	z is a n x q discrete data matrix (n is the sample size and q is the number of discrete variables).
`y`	y is a n x p continous data matrix (n is the sample size and p is the number of continous variables).
`tune1`	Tuning parameter vector for logistic regression (rho in the orginal paper).
`tune2`	Tuning parameter vector for linear regression (chi in the orginal paper).
`method`	Can only be max or min, which implies the function takes the maximum or minimum of absolute values as the final estimate.
`kappa`	tuning parameters for lambda.
`penalty1`	Penalty for logistics regression. The default penalty is weighted lasso penalty. See details at formulation (10) in High-dimensional Mixed Graphical Models.
`penalty2`	Penalty for linear regression. The default penalty is weighted lasso penalty. See details atformulation (11) in High-dimensional Mixed Graphical Models.

Details

The graph structure is estimated by maximizing the conditional likelihood of one variable given the rest. We focus on the conditional log-likelihood of each variable and fit separate regressions to estimate the parameters, much in the spirit of the neighborhood selection approach proposed by Meinshausen-Buhlmann for the Gaussian graphical model and by Ravikumar for the Ising model. We incorporating a group lasso penalty, approximated by a weighted lasso penalty for computational efficiency.

Value

The function returns is a structure of fitted parameters path, the notations are the same as the paper.

`fitlist_post`	the fitted parameter path by taking the maximum or minimum absolute values with signs
`fitlist`	The original fitlist

Author(s)

Mingyu Qi, Tianxi Li

References

Jie Cheng, Tianxi Li, Elizaveta Levina, and Ji Zhu.(2017) *High-dimensional Mixed Graphical Models. Journal of Computational and Graphical Statistics 26.2: 367-378*, https://arxiv.org/pdf/1304.2810.pdf

Simon, N., Friedman, J., Hastie,T., Tibshirani, R. (2011) *Regularization Paths for Cox's ProportionalHazards Model via Coordinate Descent, Journal of Statistical Software, Vol.39(5) 1-13*, https://www.jstatsoft.org/v39/i05/

Meinshausen, N. and Buhlmann, P. (2006) *High dimensional graphs and variable selection with the lasso, Annals of Statistics, 34, 1436–1462.*, https://arxiv.org/pdf/math/0608017.pdf

Ravikumar, P., Wainwright, M., and Lafferty, J. (2010) *High-dimensionalIsing model selection using l1-regularized logistic regression,Annals of Statistics, 38, 1287–1319.*, https://arxiv.org/pdf/1010.0311.pdf

Liu, H., Han, F., Yuan, M., Lafferty, J., and Wasserman, L. (2012) *High dimensional semiparametric Gaussian copula graphical models, Annals of Statistics, 40, 2293–2326.*, https://arxiv.org/pdf/1202.2169.pdf

Zhao, P., Rocha, G., and Yu, B. (2009) *The composite absolute penalties family for grouped and hierarchical variable selection, The Annals of Statistics, 3468–3497.*, https://arxiv.org/pdf/0909.0411.pdf

See Also

datagen

Examples

```
n = 100
p = 20
q = 10
a = 1
b=  2
c = 1


adj = matrix(0, p+q, p+q)
adj[10:16, 10:16] = 1
adj[1:5, 1:5] = 1
adj[25:30, 25:30] = 1
adj = adj-diag(diag(adj))

parlist = pargen(adj, p, q, a, b,c)

mydata = datagen(parlist, n)

z = mydata$z

y = mydata$y

tune1 = tune2 = 0.1

kappa = 0.1

## parameter estimation

fit = hmgm(z, y, tune1, tune2, 'max', kappa)
```

---

pargen *Generating parameters according to the graph*

Description

The function generates parameters for different types of edges based on the graph.

Usage

```
pargen(adjmat, p, q, a, b, c)
```

Arguments

`adjmat`	A m x m adjacency matrix (m is the number of total variables). The program automatically check whether the matrix is symmetric and positive.
`p`	The number of continous variables.
`q`	The number of binary variables.
`a`	Control overall magnitude of the non-zero parameters for edges connecting continuous variables.
`b`	Control overall magnitude of the non-zero parameters for edges connecting binary and continuous variables.
`c`	Control overall magnitude of the non-zero parameters for edges connecting binary variables.

Details

In order to generate simulation data, first generate the parameters. Once the adjacency matrix is given, we set all parameters corresponding to absent edges to 0. For the non-zero parameters, we set `lambdaj`, `lambdajk`, `etaj` to be positive or negative with equal probability and the absolute value of each non-zero `etaj` is drawn from the uniform distribution on the interval (0.9a, 1.1a) and each non-zero `lambdaj` or `lambdajk` is from (0.9c,1.1c). The program makes sure that all the probability values are not negative.

Value

The function returns a paramter list.

Author(s)

Mingyu Qi, Tianxi Li

References

Jie Cheng, Tianxi Li, Elizaveta Levina, and Ji Zhu. (2017) *High-dimensional Mixed Graphical Models. Journal of Computational and Graphical Statistics 26.2: 367-378*, https://arxiv.org/pdf/1304.2810.pdf

See Also

`datagen`

Examples

```
## set controlling parameters
p = 20
q = 10
a = 1
b = 2
c = 1

# set adjacency matrix
adj = matrix(0, p+q, p+q)
adj[10:16, 10:16] = 1
adj[1:5, 1:5] = 1
adj[25:30, 25:30] = 1
adj = adj-diag(diag(adj))

#generate list
parlist = pargen(adj, p, q, a, b,c)
```

---

pargroup *Function to partition overlapping groups into non-overlapping groups*

Description

Function to identify all regions where groups intersect, make all variables in each overlapping region into a new group.

Usage

```
pargroup(A)
```

Arguments

A	An n x p matrix represents the relationship between variables and groups. (n is the number of groups and p is the number of variables)

Details

In order to partition groups, we propose a method based on Gaussian-Jordan elimination with pivot on `A` to get a reduced row echelon form matrix. Then we use the reduced row echelon form matrix to determine groups. This method can obtain an accurate result as well as reduce computational complexity in R.

Value

A m x p matrix which represents the relationship between variables and groups after partitioning.

Author(s)

Mingyu Qi, Tianxi Li

References

Jie Cheng, Tianxi Li, Elizaveta Levina, and Ji Zhu. (2017) *High-dimensional Mixed Graphical Models. Journal of Computational and Graphical Statistics 26.2: 367-378*, https://arxiv.org/pdf/1304.2810.pdf

Examples

```
## Set an overlap group
A<-rbind(c(1,1,1,0,0), c(0,1,1,1,1))


## Use pargroup() to partion this overlap group to non-overlap group

G = pargroup(A)
```

Index