### Package ‘fastclime’

April 29, 2015

**Type** Package  
**Title** A Fast Solver for Parameterized Linear Programming Problems and Constrained L1 Minimization Approach to Sparse Precision Matrix Estimation  
**Version** 1.2.5  
**Date** 2015-04-26  
**Author** Haotian Pang, Han Liu and Robert Vanderbei  
**Maintainer** Haotian Pang <hpang@princeton.edu>  
**Depends** R (>= 2.15.0), lattice, igraph, MASS, Matrix  
**Description** An efficient method of recovering precision matrices by applying the parametric simplex method is provided in this package. The computation is based on a linear optimization solver. It also contains a generic Linear Programming solver and a parameterized Linear Programming solver based on the parametric simplex method.  
**License** GPL-2  
**Repository** CRAN  
**NeedsCompilation** yes  
**Date/Publication** 2015-04-29 01:00:45

#### R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>fastclime-package</td>
<td>2</td>
</tr>
<tr>
<td>fastclime</td>
<td>2</td>
</tr>
<tr>
<td>fastclime.generator</td>
<td>4</td>
</tr>
<tr>
<td>fastclime.lambda</td>
<td>6</td>
</tr>
<tr>
<td>fastclime.plot</td>
<td>8</td>
</tr>
<tr>
<td>fastlp</td>
<td>9</td>
</tr>
<tr>
<td>paralp</td>
<td>10</td>
</tr>
<tr>
<td>plot.fastclime</td>
<td>11</td>
</tr>
<tr>
<td>plot.sim</td>
<td>12</td>
</tr>
<tr>
<td>print.fastclime</td>
<td>12</td>
</tr>
<tr>
<td>print.sim</td>
<td>13</td>
</tr>
<tr>
<td>stockdata</td>
<td>14</td>
</tr>
</tbody>
</table>
Index

fastclime-package  Fast Parametric Simplex Solver for CLIME and Linear Programming

Description

A package for generic linear programming, parameterized linear programming and constrained l1 minimization approach to sparse precision matrix estimation

Details

Package: fastclime
Type: Package
Version: 1.2.4
Date: 2014-04-25
License: GPL-2
LazyLoad: yes

The package "fastclime" provides 4 main functions:
(1) the data generator creates random samples from multivariate normal distributions with different graph structures. Please refer to fastclime.generator.
(2) The parametric simplex solver for constrained l1 minimization approach to sparse precision matrix estimation. Please refer to fastclime.
(3) The path selector function gives the path and precision matrix for a given parameter in CLIME. Please refer to fastclime.lambda.
(4) A generic linear programming solver and a parameterized linear programming solver. Please refer to fastlp and paralp.

Author(s)

Haotian Pang, Han Liu and Robert Vanderbei
Maintainer: Haotan Pang<hpang@princeton.edu>

See Also

fastclime.generator, fastclime, fastclime.plot, fastclime.lambda, fastlp and paralp

Description

A fast parametric simplex solver for constrained l1 minimization approach to sparse precision matrix estimation
Usage

```r
fastclime(x, lambda.min = 0.1, nlambda = 50)
```

Arguments

- `x` There are 2 options: (1) `x` is an \( n \) by \( d \) data matrix (2) a \( d \) by \( d \) sample covariance matrix. The program automatically identifies the input matrix by checking the symmetry. (\( n \) is the sample size and \( d \) is the dimension)
- `lambda.min` This is the smallest value of lambda you would like the solver to explorer. The default value is \( 0.1 \). If `nlambda` is large enough, the precision matrix selector function `fastclime.lambda` will be able to find all precision matrix corresponding to all lambda values ranging from 1 to `lambda.min`.
- `nlambda` It is the number of the path length one would like to achieve. The default length is 50. Note if \( d \) is large and `nlambda` is also large, it is possible that the program will fail to allocate memory for the path.

Details

This program uses parametric simplex linear programming method to solve CLIME (Constrained l1 Minimization Sparse Precision Matrix Estimation) problem. The solution path of the problem corresponds to the parameter in the parametric simplex method.

Value

An object with S3 class "fastclime" is returned:

- `data` The \( n \) by \( d \) data matrix or \( d \) by \( d \) sample covariance matrix from the input
- `cov.input` An indicator of the sample covariance.
- `sigmahat` The empirical covariance of the data. If cov.input is TRUE, sigmahat = data
- `maxnlambda` The length of the path. If the program finds `lambda.min` in less than `nlambda` iterations for all columns, then the actual maximum length for all columns will be returned. Otherwise it equals `nlambda`.
- `lambdamtx` The sequence of regularization parameters for each column, it is a `nlambda` by \( d \) matrix. It will be filled with 0 when the program finds the required `lambda.min` value for that column. This parameter is required for `fastclime.lambda`.
- `icovlist` A `nlambda` list of \( d \) by \( d \) precision matrices as an alternative graph path (numerical path) corresponding to `lambdamtx`. This parameter is also required for `fastclime.lambda`.

Note

The program will stop when either the maximum number of iteration for each column `nlambda` is achieved or when the required `lambda.min` is achieved for each column. When the dimension is huge, make sure `nlambda` is small so that there are enough memory to allocate the solution path. `lambdamtx` and `icovlist` will be used in `fastclime.lambda`. 


Author(s)
Haotian Pang, Han Liu and Robert Vanderbei
Maintainer: Haotan Pang<hpang@princeton.edu>

See Also
fastclime.generator.fastclime.plot, fastclime.lambda and fastclime-package.

Examples
#generate data
L = fastclime.generator(n = 100, d = 20)

#graph path estimation
out1 = fastclime(L$data, 0.1)
O = fastclime.lambda(out1$lambdamtx, out1$icovlist, 0.2)
fastclime.plot(O$path)

#graph path estimation using the sample covariance matrix as the input.
out1 = fastclime(cor(L$data), 0.1)
O = fastclime.lambda(out1$lambdamtx, out1$icovlist, 0.2)
fastclime.plot(O$path)

Description

Implements the data generation from multivariate normal distributions with different graph structures, including "random", "hub", "cluster" and "band".

Usage
fastclime.generator(n = 200, d = 50, graph = "random", v = NULL, u = NULL, 
g = NULL, prob = NULL, vis = FALSE, verbose = TRUE)

Arguments

n
The number of observations (sample size). The default value is 200.

d
The number of variables (dimension). The default value is 50.

graph
The graph structure with 4 options: "random", "hub", "cluster" and "band".

v
The off-diagonal elements of the precision matrix, controlling the magnitude of partial correlations with u. The default value is 0.3.

u
A positive number being added to the diagonal elements of the precision matrix, to control the magnitude of partial correlations. The default value is 0.1.
fastclime.generator

5

For "cluster" or "hub" graph, g is the number of hubs or clusters in the graph. The default value is about \( \frac{d}{20} \) if \( d \geq 40 \) and 2 if \( d < 40 \). For "band" graph, \( g \) is the bandwidth and the default value is 1. NOT applicable to "random" graph.

prob

For "random" graph, it is the probability that a pair of nodes has an edge. The default value is \( \frac{6}{d} \) if \( d \) is greater than or equal to 30 and \( \frac{1}{3} \) if \( d < 30 \). NOT applicable to "hub" or "band" graphs.

vis

For "random" graph, it is the probability that a pair of nodes has an edge. The default value is \( \frac{6}{d} \) if \( d \) is greater than or equal to 30 and \( \frac{1}{3} \) if \( d < 30 \). NOT applicable to "hub" or "band" graphs.

visualize the adjacency matrix of the true graph structure, the graph pattern, the covariance matrix and the empirical covariance matrix. The default value is FALSE.

verbose

If verbose = FALSE, tracing information printing is disabled. The default value is TRUE.

Details

Given the adjacency matrix \( \theta \), the graph patterns are generated as below:

(I) "random": Each pair of off-diagonal elements are randomly set \( \theta[i,j] = 1 \) for \( i \neq j \) with probability prob, and 0 otherwise. It results in about \( d \times (d-1) \times \text{prob} / 2 \) edges in the graph.

(II) "hub": The row/columns are evenly partitioned into \( g \) disjoint groups. Each group is associated with a "center" row \( i \) in that group. Each pair of off-diagonal elements are set \( \theta[i,j] = 1 \) for \( i \neq j \) if \( j \) also belongs to the same group as \( i \) and 0 otherwise. It results in \( d - g \) edges in the graph.

(III) "cluster": The row/columns are evenly partitioned into \( g \) disjoint groups. Each pair of off-diagonal elements are set \( \theta[i,j] = 1 \) for \( i \neq j \) if both \( i \) and \( j \) belong to the same group, and 0 otherwise. It results in about \( g \times (d/g) \times (d/g-1) \times \text{prob} / 2 \) edges in the graph.

(IV) "band": The off-diagonal elements are set to be \( \theta[i,j] = 1 \) if \( 1 \leq |i-j| \leq g \) and 0 otherwise. It results in \( (2d-1-g) \times g / 2 \) edges in the graph.

The adjacency matrix \( \theta \) has all diagonal elements equal to 0. To obtain a positive definite precision matrix, the smallest eigenvalue of \( \theta + v \) (denoted by \( e \)) is computed. Then we set the precision matrix equal to \( \theta + v + (|e| + 0.1 + u)I \). The covariance matrix is then computed to generate multivariate normal data.

Value

An object with S3 class "sim" is returned:

data The n by d matrix for the generated data
sigma The covariance matrix for the generated data
omega The precision matrix for the generated data
select the precision matrix and solution path for a given parameter lambda

Usage

fastclime.lambda(lambdamtx, icovlist, lambda)
**Arguments**

- `lambdamtx`: The sequence of regularization parameters for each column, it is a nlambda by d matrix.
- `icovlist`: A nlambda list of d by d precision matrices as an alternative graph path (numerical path) corresponding to `lambdamtx`.
- `lambda`: The user specified parameter lambda. The function will return the solution path corresponding to this value. Note lambda has to be larger than or equal to lambda.min input in fastclime.

**Details**

The output from fastclime stores a list of precision matrices and a matrix of parameters. This program will select the required solution path and precision matrix for a given parameter lambda.

**Value**

An object with S3 class "fastclime.lambda" is returned:

- `icov`: The estimated precision matrix corresponding to lambda.
- `path`: The estimated graph path corresponding to lambda.
- `sparsity`: The sparsity level of this estimated graph path.

**Note**

The function is able to estimate the precision matrices corresponding to all lambda values ranging from 1 to lambda.min, provided a large enough nlambda is used in `fastclime`. The function will give a message if the program could not find the solution path corresponding to the required lambda. The user may want to increase nlambda in `fastclime` in order to find the required solution path.

**Author(s)**

Haotian Pang, Han Liu and Robert Vanderbei
Maintainer: Haotan Pang<hpang@princeton.edu>

**See Also**

`fastclime` and `fastclime-package`

**Examples**

```r
#generate data
L = fastclime.generator(n = 100, d = 20)

#graph path estimation
out1 = fastclime(L$data, 0.1)  
O = fastclime.lambda(out1$lambdamtx, out1$icovlist, 0.2)  
fastclime.plot(O$path)
```
**fastclime.plot**

*Graph visualization*

**Description**

Implements the graph visualization using adjacency matrix. It can automatic organize 2D embedding layout.

**Usage**

```r
class.plot(G, epsflag = FALSE, graph.name = "default", cur.num = 1, location)
```

**Arguments**

- `G`: The adjacency matrix corresponding to the graph.
- `epsflag`: If `epsflag = TRUE`, save the plot as an eps file in the target directory. The default value is `FALSE`.
- `graph.name`: The name of the output eps files. The default value is "default".
- `cur.num`: The number of plots saved as eps files. Only applicable when `epsflag = TRUE`. The default value is 1.
- `location`: Target directory. The default value is the current working directory.

**Details**

The user can change `cur.num` to plot several figures and select the best one. The implementation is based on the popular package "igraph".

**Author(s)**

Haotian Pang, Han Liu and Robert Vanderbei

Maintainer: Haotan Pang <hpang@princeton.edu>

**See Also**

- `fastclime`
- `fastclime-package`

**Examples**

```r
## visualize the hub graph
L = fastclime.generator(graph = "hub")
fastclime.plot(L$theta)

## visualize the band graph
L = fastclime.generator(graph = "band", g=5)
fastclime.plot(L$theta)
```
## visualize the cluster graph
```r
L = fastclime.generator(graph = "cluster")
fastclime.plot(L$theta)
```

# show working directory
getwd()

# plot 5 graphs and save the plots as eps files in the working directory
```r
fastclime.plot(L$theta, epsflag = TRUE, cur.num = 5)
```

---

### fastlp

**A generic LP solver**

#### Description

A generic linear programming solver using parametric simplex method

#### Usage

```r
fastlp(obj, mat, rhs, lambda=0)
```

#### Arguments

- `obj`: The objective vector of the coefficient with length n.
- `mat`: The constraint matrix of the linear programming with dimension m*n. Note this argument must be in matrix form even it is a vector.
- `rhs`: The right hand side vector of the constraint with length m.
- `lambda`: The parametric simplex method will stop when the calculated parameter is smaller than lambda. The default value is zero and it corresponds to the optimal value.

#### Details

This function is used to solve a general linear programming in standard inequality form: "maximize obj*x, subject to: mat*x<=rhs, x>=0"

#### Value

The optimal value will be returned if it exists. Otherwise the function will indicate the problem is infeasible or unbounded.

#### Note

The linear programming should be in the form "maximize obj*x, subject to: mat*x<=rhs, x>=0". If the original problem is not in this form. The user has to convert it into this form. For example, the equality constraints can be separated into two inequality constraints.
Author(s)
Haotian Pang, Han Liu and Robert Vanderbei
Maintainer: Haotan Pang <hpang@princeton.edu>

See Also
fastclime and fastclime-package

Examples
# generate an LP problem and solve it
A = matrix(c(-1, -1, 0, 1, -2, 1), nrow = 3)
b = c(-1, -2, 1)
c = c(-2, 3)
fastlp(c, A, b)

paralp A solver for parameterized LP problems

Description
A parameterized linear programming solver using parametric simplex method

Usage
paralp(obj, mat, rhs, obj_bar, rhs_bar, lambda = 0)

Arguments
  obj     The objective vector of the coefficient with length n.
  mat     The constraint matrix of the linear programming with dimension m*n. Note this argument must be in matrix form even it is a vector.
  rhs     The right hand side vector of the constraint with length m.
  obj_bar The vector used to time the parameter and added to the objective vector, with length n. This perturbation vector must be nonnegative.
  rhs_bar The vector used to time the parameter and added to the right hand side vector, with length m. This perturbation vector must be nonnegative.
  lambda  The parametric simplex method will stop when the calculated parameter is smaller than lambda. The default value is zero and it corresponds to the optimal value.

Details
This function is used to solve a general linear programming in standard inequality form: "maximize obj'*x + obj_bar'*lambda, subject to: mat'*x <= rhs + rhs_bar'*lambda, x >= 0"
Value

The optimal value will be returned if it exists with a proper value of chosen lambda. Otherwise the function will indicate the problem is infeasible or unbounded.

Author(s)

Haotian Pang, Han Liu and Robert Vanderbei
Maintainer: Haotan Pang<hpang@princeton.edu>

See Also

fastclime and fastclime-package

Examples

# generate an LP problem and solve it
A = matrix(c(-1L, -1L, 0L, 1L, -2L, 1L), nrow=3)
b = c(-1L, -2L, 1L)
c = c(-2L, 3L)
b_bar = c(1L, 1L)
c_bar = c(1L, 1L)
paralp(c, A, b, c_bar, b_bar)

plot.fastclime  

Plot function for S3 class "fastclime"

Description

Plot sparsity level information (the first column) from the graph path

Usage

## S3 method for class 'fastclime'
plot(x, ...)

Arguments

x  
An object with S3 class "fastclime"

...  
System reserved (No specific usage)

Author(s)

Haotian Pang, Han Liu and Robert Vanderbei
Maintainer: Haotan Pang<hpang@princeton.edu>

See Also

fastclime
plot.sim  

Plot function for S3 class "sim"

Description

Visualize the covariance matrix, the empirical covariance matrix, the adjacency matrix and the graph pattern of the true graph structure.

Usage

## S3 method for class 'sim'
plot(x, ...)

Arguments

x An object with S3 class "sim"

... System reserved (No specific usage)

Author(s)

Haotian Pang, Han Liu and Robert Vanderbei
Maintainer: Haotan Pang<hpang@princeton.edu>

See Also

fastclime.generator and fastclime

print.fastclime  

Print function for S3 class "fastclime"

Description

Print the information about the model usage, the graph path length, graph dimension, sparsity level.

Usage

## S3 method for class 'fastclime'
print(x, ...)

Arguments

x An object with S3 class "fastclime"

... System reserved (No specific usage)
print.sim

Author(s)

Haotian Pang, Han Liu and Robert Vanderbei
Maintainer: Haotan Pang hpang@princeton.edu

See Also

fastclime and fastclime

print.sim

Print function for S3 class "sim"

Description

Print the information about the sample size, the dimension, the pattern and sparsity of the true graph structure.

Usage

## S3 method for class 'sim'
print(x, ...)

Arguments

x An object with S3 class "sim"
...

System reserved (No specific usage)

Author(s)

Haotian Pang, Han Liu and Robert Vanderbei
Maintainer: Haotan Pang hpang@princeton.edu

See Also

fastclime.generator and fastclime.generator
Description

This data set consists of stock price and company information.

Usage

data(stockdata)

Format

The format is a list containing two matrices. 1. data - 1258x452, represents the 452 stocks’ close prices for 1258 trading days. 2. info - 452x3: The 1st column: the query symbol for each company. The 2nd column: the category for each company. The 3rd column: the full name of each company.

Details

This data set can be used to perform high-dimensional graph estimation to analyze the relationships between S&P 500 companies.

Author(s)

Haotian Pang, Han Liu and Robert Vanderbei
Maintainer: Haotan Pang<hpang@princeton.edu>

Source

It is publicly available at http://ichart.finance.yahoo.com

Examples

data(stockdata)
image(stockdata$data)
stockdata$info
Index

*Topic datasets
  stockdata, 14

fastclime, 2, 2, 6–8, 10–13
fastclime-package, 2
fastclime.generator, 2, 4, 4, 12, 13
fastclime.lambda, 2–4, 6
fastclime.plot, 2, 4, 8
fastlp, 2, 9

paralp, 2, 10
plot.fastclime, 11
plot.sim, 12
print.fastclime, 12
print.sim, 13

stockdata, 14