Package ‘clime’

April 9, 2010

Type Package

Title Constrained L1-minimization for Inverse (covariance) Matrix Estimation

Version 0.2

Date 2010-04-09

Author T. Tony Cai, Weidong Liu and Xi (Rossi) Luo

Maintainer Xi (Rossi) Luo <xi.rossi.luo@gmail.com>

Depends lpSolve

Description A robust constrained L1 minimization method for estimating a large sparse inverse covariance matrix (aka precision matrix), and recovering its support for building graphical models. The computation uses linear programming.

License GPL-2

R topics documented:

clime *solve for the inverse matrix*

Description

Solve for a series of the inverse covariance matrix estimates at a grid of values for the constraint lambda.

Usage

```
clime(x, lambda=NULL, nlambda=ifelse(is.null(lambda),100,length(lambda)),
      lambda.max=0.8, lambda.min=ifelse(nrow(x)>ncol(x), 1e-4, 1e-2),
      sigma=FALSE, perturb=FALSE, standardize=TRUE, logspaced=TRUE,
      linsolver=c("primaldual", "simplex"), pdtol=1e-3, pdmaxiter=50)
```

Arguments

`x`	Input matrix of size n (observations) times p (variables). Each column is a variable of length n. Alternatively, the sample covariance matrix may be set here with the next option `sigma` set to be TRUE. When the input is the sample covariance matrix, `cv.clime` can not be used for this object.
`lambda`	Grid of non-negative values for the constraint parameter lambda. If missing, `nlambda` values from `lambda.min` to `lambda.max` will be generated.
`standardize`	Whether the variables will be standardized to have mean zero and unit standard deviation. Default TRUE.
`nlambda`	Number of values for program generated `lambda`. Default 100.
`lambda.max`	Maximum value of program generated `lambda`. Default 0.8.
`lambda.min`	Minimum value of program generated `lambda`. Default 1e-4($n > p$) or 1e-2($n < p$).
`sigma`	Whether `x` is the sample covariance matrix. Default FALSE.
`perturb`	Whether a perturbed `Sigma` should be used or the positive perturbation added if it is numerical. Default FALSE.
`logspaced`	Whether program generated lambda should be log-spaced or linear spaced. Default TRUE.
`linsolver`	Whether `primaldual` (default) or `simplex` method should be employed. Rule of thumb: `primaldual` for large p, `simplex` for small p.
`pdtol`	Tolerance for the duality gap, ignored if `simplex` is employed.
`pdmaxiter`	Maximum number of iterations for `primaldual`, ignored if `simplex` is employed.

Details

A constrained ℓ_1 minimization approach for sparse precision matrix estimation (details in references) is implemented here using linear programming (revised simplex or primal-dual interior point method). It solves a sequence of `lambda` values on the following objective function

$$\min |\Omega|_1 \quad \text{subject to: } ||\Sigma_n \Omega - I||_\infty \le \lambda$$

where Σ_n is the sample covariance matrix and Ω is the inverse we want to estimate.

Value

An object with S3 class "`clime`". You can also use it as a regular R list with the following fields:

`Omega`	List of estimated inverse covariance matrix for a grid of values for `lambda`.
`lambda`	Actual sequence of `lambda` used in the program
`perturb`	Actual perturbation used in the program.
`standardize`	Whether standardization is applied to the columns of `x`.
`x`	Actual `x` used in the program.
`lpfun`	Linear programming solver used.

Author(s)

T. Tony Cai, Weidong Liu and Xi (Rossi) Luo
Maintainer: Xi (Rossi) Luo <xi.rossi.luo@gmail.com>

References

Cai, T.T., Liu, W., and Luo, X. (2010). *A constrained ℓ_1 minimization approach for sparse precision matrix estimation.* Technical Report, University of Pennsylvania.

Examples

```
## trivial example
n <- 50
p <- 5
X <- matrix(rnorm(n*p), nrow=n)
re.clime <- clime(X)

## tridiagonal matrix example
bandMat <- function(p, k) {
  cM <- matrix(rep(1:p, each=p), nrow=p, ncol=p)
  return((abs(t(cM)-cM)<=k)*1)
}
## tridiagonal Omega with diagonal 1 and off-diagonal 0.5
Omega <- bandMat(p, 1)*0.5
diag(Omega) <- 1
Sigma <- solve(Omega)
X <- matrix(rnorm(n*p), nrow=n)%*%chol(Sigma)
re.clime <- clime(X, standardize=FALSE, linsolver="simplex")
```

```
re.cv <- cv.clime(re.clime)
re.clime.opt <- clime(X, standardize=FALSE, re.cv$lambdaopt)

## Compare Frobenius norm loss
## clime estimator
sqrt( sum( (Omega-re.clime.opt$Omegalist[[1]])^2 ) )
## Not run: 0.3438533
## Sample covariance matrix inversed
sqrt( sum( ( Omega-solve(cov(X)*(1-1/n)) )^2 ) )
## Not run: 0.874041
sqrt( sum( ( Omega-solve(cov(X)) )^2 ) )
## Not run: 0.8224296
```

`clime-internal` *internal clime functions*

Description

Internal clime functions

Usage

```
likelihood(Sigma, Omega)
tracel2(Sigma, Omega)
```

Arguments

`Sigma`	Covariance matrix.
`Omega`	Inverse covariance matrix.

Details

There are not intended for use by users.

Author(s)

T. Tony Cai, Weidong Liu and Xi (Rossi) Luo
Maintainer: Xi (Rossi) Luo <xi.rossi.luo@gmail.com>

References

Cai, T.T., Liu, W., and Luo, X. (2010). *A constrained ℓ_1 minimization approach for sparse precision matrix estimation.* Technical Report, University of Pennsylvania.

cv.clime — *k-fold cross validation for clime object*

Description

Perform a k-fold cross validation for selecting lambda

Usage

```
cv.clime(clime.obj, loss=c("likelihood", "tracel2"), fold=5)
```

Arguments

`clime.obj`	clime object output from `clime`. Note that this requires that the input to `clime` is `x` instead of the sample covariance matrix.
`loss`	loss to be used in cross validation. Currently, two losses are available: "likelihood" and "tracel2". Default "likelihood".
`fold`	number of folds used in cross validation. Default 5.

Details

Perform a k-fold cross validation for selecting the tuning parameter `lambda` in clime. Two losses are implemented currently:

$$\text{likelihood: } Tr[\Sigma\Omega] - \log|\Omega| - p$$

$$\text{tracel2: } Tr[diag(\Sigma\Omega - I)^2].$$

Value

An object with S3 class `"cv.clime"`. You can use it as a regular R list with the following fields:

`lambdaopt`	the lambda selected by cross validation to minimize the loss over the grid values of `lambda`.
`loss`	the name of loss used in cross validation.
`lambda`	sequence of `lambda` used in the program.
`loss.mean`	average k-fold loss values for each grid value `lambda`.
`loss.mean`	standard deviation of k-fold loss values for each grid value `lambda`.
`lpfun`	Linear programming solver used.

Author(s)

T. Tony Cai, Weidong Liu and Xi (Rossi) Luo
Maintainer: Xi (Rossi) Luo `<xi.rossi.luo@gmail.com>`

References

Cai, T.T., Liu, W., and Luo, X. (2010). *A constrained ℓ_1 minimization approach for sparse precision matrix estimation.* Technical Report, University of Pennsylvania.

Examples

```
## trivial example
n <- 50
p <- 5
X <- matrix(rnorm(n*p), nrow=n)
re.clime <- clime(X)
re.cv <- cv.clime(re.clime)
re.clime.opt <- clime(X, re.cv$lambdaopt)
```

---

print.clime *print a clime object*

Description

Print a summary of the clime object.

Usage

```
## S3 method for class 'clime':
print(x,digits = max(3, getOption("digits") - 3), ... )
```

Arguments

x	clime object.
digits	significant digits in printout.
...	additional print options.

Details

This call simply outlines the options used for computing a clime object.

Value

The output above is invisiblly returned.

Author(s)

T. Tony Cai, Weidong Liu and Xi (Rossi) Luo
Maintainer: Xi (Rossi) Luo <xi.rossi.luo@gmail.com>

References

Cai, T.T., Liu, W., and Luo, X. (2010). *A constrained ℓ_1 minimization approach for sparse precision matrix estimation.* Technical Report, University of Pennsylvania.

Examples

```
## trivial example
n <- 50
p <- 5
X <- matrix(rnorm(n*p), nrow=n)
re.clime <- clime(X)
print(re.clime)
```

`print.cv.clime` *print a cross validated clime object*

Description

Print a summary of the cv.clime object.

Usage

```
## S3 method for class 'cv.clime':
print(x,digits = max(3, getOption("digits") - 3), ... )
```

Arguments

`x`	cv.clime object.
`digits`	significant digits in printout.
`...`	additional print options.

Details

This call outputs first a three column matrix with `lambda`, `mean` and `sd` for the cross validation loss values. The actual loss used and the optimal `lambda` value picked by cv are printed.

Value

The output above is invisiblly returned.

Author(s)

T. Tony Cai, Weidong Liu and Xi (Rossi) Luo
Maintainer: Xi (Rossi) Luo `<xi.rossi.luo@gmail.com>`

References

Cai, T.T., Liu, W., and Luo, X. (2010). *A constrained ℓ_1 minimization approach for sparse precision matrix estimation.* Technical Report, University of Pennsylvania.

Examples

```
## trivial example
n <- 50
p <- 5
X <- matrix(rnorm(n*p), nrow=n)
re.clime <- clime(X)
re.cv <- cv.clime(re.clime)
print(re.cv)
```

Index