Package ‘MRCE’

October 12, 2022

Type Package
Title Multivariate Regression with Covariance Estimation
Version 2.4
Date 2022-01-04
Author Adam J. Rothman
Maintainer Adam J. Rothman <arothman@umn.edu>
Depends R (>= 2.10.1), glasso
Description Compute and select tuning parameters for the MRCE estimator proposed by Rothman, Levina, and Zhu (2010) <doi:10.1198/jcgs.2010.09188>. This estimator fits the multiple output linear regression model with a sparse estimator of the error precision matrix and a sparse estimator of the regression coefficient matrix.
License GPL-2
NeedsCompilation yes
Repository CRAN
Date/Publication 2022-01-04 17:30:10 UTC

R topics documented:

MRCE-package .................................................. 1
mrce .......................................................... 2
stock04 ......................................................... 6

Index

MRCE-package Multivariate regression with covariance estimation

Description

Computes the MRCE estimators (Rothman, Levina, and Zhu, 2010) and has the dataset stock04 used in Rothman, Levina, and Zhu (2010), originally analyzed in Yuan et al. (2007).
Details

The primary function is mrce. The dataset is stock04.

Author(s)

Adam J. Rothman

Maintainer: Adam J. Rothman <arothman@umn.edu>

References


mrce

Do multivariate regression with covariance estimation (MRCE)

Description

Let $S_q^n$ be the set of $q$ by $q$ symmetric and positive definite matrices and let $y_i \in R^q$ be the measurements of the $q$ responses for the $i$th subject ($i = 1, \ldots, n$). The model assumes that $y_i$ is a realization of the $q$-variate random vector

$$Y_i = \mu + \beta' x_i + \varepsilon_i, \quad i = 1, \ldots, n$$

where $\mu \in R^q$ is an unknown intercept vector; $\beta \in R^{p \times q}$ is an unknown regression coefficient matrix; $x_i \in R^p$ is the known vector of values for $i$th subject’s predictors, and $\varepsilon_1, \ldots, \varepsilon_n$ are $n$ independent copies of a $q$-variate Normal random vector with mean 0 and unknown inverse covariance matrix $\Omega \in S_q^n$.

This function computes penalized likelihood estimates of the unknown parameters $\mu$, $\beta$, and $\Omega$. Let

$$\bar{y} = n^{-1} \sum_{i=1}^n y_i \quad \text{and} \quad \bar{x} = n^{-1} \sum_{i=1}^n x_i.$$

These estimates are

$$\hat{(\beta, \Omega)} = \arg \min_{(B,Q) \in R^{p \times q} \times S_q^n} \left\{ g(B, Q) + \lambda_1 \left( \sum_{j \neq k} |Q_{jk}| + 1(p \geq n) \sum_{j=1}^q |Q_{jj}| \right) + 2\lambda_2 \sum_{j=1}^p \sum_{k=1}^q |B_{jk}| \right\}$$

and

$$\hat{\mu} = \bar{y} - \hat{\beta}' \bar{x},$$

where

$$g(B, Q) = \text{tr} \{ n^{-1}(Y - XB)'(Y - XB)Q \} - \log |Q|,$$

$Y \in R^{n \times q}$ has $i$th row $(y_i - \bar{y})'$, and $X \in R^{n \times p}$ has $i$th row $(x_i - \bar{x})'.$
Usage

mrce(X, Y, lam1=NULL, lam2=NULL, lam1.vec=NULL, lam2.vec=NULL,
method=c("single", "cv", "fixed.omega"),
cov.tol=1e-4, cov.maxit=1e3, omega=NULL,
maxit.out=1e3, maxit.in=1e3, tol.out=1e-8,
tol.in=1e-8, kfold=5, silent=TRUE, eps=1e-5,
standardize=FALSE, permute=FALSE)

Arguments

X
   An n by p matrix of the values for the prediction variables. The ith row of X is
   xi defined above (i = 1,...,n). Do not include a column of ones.

Y
   An n by q matrix of the observed responses. The ith row of Y is yi defined above
   (i = 1,...,n).

lam1
   A single value for λ1 defined above. This argument is only used if method="single"

lam2
   A single value for λ2 defined above (or a p by q matrix with (j,k)th entry λ2jk in
   which case the penalty 2λ2 ∑j=1p ∑k=1q |Bjk| becomes
   2 ∑j=1p ∑k=1q λ2jk |Bjk|).
   This argument is not used if method="cv".

lam1.vec
   A vector of candidate values for λ1 from which the cross validation procedure
   searches: only used when method="cv" and must be specified by the user when
   method="cv". Please arrange in decreasing order.

lam2.vec
   A vector of candidate values for λ2 from which the cross validation procedure
   searches: only used when method="cv" and must be specified by the user when
   method="cv". Please arrange in decreasing order.

method
   There are three options:
   • method="single" computes the MRCE estimate of the regression coefficient matrix with penalty tuning parameters lam1 and lam2;
   • method="cv" performs kfold cross validation using candidate tuning parameters in lam1.vec and lam2.vec;
   • method="fixed.omega" computes the regression coefficient matrix estimate for which Q (defined above) is fixed at omega.

cov.tol
   Convergence tolerance for the glasso algorithm that minimizes the objective function (defined above) with B fixed.

cov.maxit
   The maximum number of iterations allowed for the glasso algorithm that minimizes the objective function (defined above) with B fixed.

omega
   A user-supplied fixed value of Q. Only used when method="fixed.omega" in
   which case the minimizer of the objective function (defined above) with Q fixed
   at omega is returned.

maxit.out
   The maximum number of iterations allowed for the outer loop of the exact
   MRCE algorithm.

maxit.in
   The maximum number of iterations allowed for the algorithm that minimizes
   the objective function, defined above, with Ω fixed.

tol.out
   Convergence tolerance for outer loop of the exact MRCE algorithm.
tol.in
Convergence tolerance for the algorithm that minimizes the objective function, defined above, with $\Omega$ fixed.

kfold
The number of folds to use when method="cv".

silent
Logical: when silent=FALSE this function displays progress updates to the screen.

eps
The algorithm will terminate if the minimum diagonal entry of the current iterate’s residual sample covariance is less than $\text{eps}$. This may need adjustment depending on the scales of the variables.

standardize
Logical: should the columns of $X$ be standardized so each has unit length and zero average. The parameter estimates are returned on the original unstandardized scale. The default is FALSE.

permute
Logical: when method="cv", should the subject indices be permuted? The default is FALSE.

Details
Please see Rothman, Levina, and Zhu (2010) for more information on the algorithm and model.
This version of the software uses the glasso algorithm (Friedman et al., 2008) through the R package glasso. If the algorithm is running slowly, track its progress with silent=FALSE. In some cases, choosing cov.tol=0.1 and tol.out=1e-10 allows the algorithm to make faster progress. If one uses a matrix for lam2, consider setting tol.in=1e-12.

When $p \geq n$, the diagonal of the optimization variable corresponding to the inverse covariance matrix of the error is penalized. Without diagonal penalization, if there exists a $\bar{B}$ such that the $q$th column of $Y$ is equal to the $q$th column of $\bar{X}\bar{B}$, then a global minimizer of the objective function (defined above) does not exist.

The algorithm that minimizes the objective function, defined above, with $Q$ fixed uses a similar update strategy and termination criterion to those used by Friedman et al. (2010) in the corresponding R package glmnet.

Value
A list containing

\begin{itemize}
  \item $\hat{B}$ This is $\hat{B} \in \mathbb{R}^{p \times q}$ defined above. If method="cv", then best.lam1 and best.lam2 defined below are used for $\lambda_1$ and $\lambda_2$.
  \item $\hat{\mu}$ This is the intercept estimate $\hat{\mu} \in \mathbb{R}^q$ defined above. If method="cv", then best.lam1 and best.lam2 defined below are used for $\lambda_1$ and $\lambda_2$.
  \item $\hat{\Omega}$ This is $\hat{\Omega} \in S^{q+}$ defined above. If method="cv", then best.lam1 and best.lam2 defined below are used for $\lambda_1$ and $\lambda_2$.
  \item $\bar{x}$ This is $\bar{x} \in \mathbb{R}^p$ defined above.
  \item $\bar{y}$ This is $\bar{y} \in \mathbb{R}^q$ defined above.
  \item best.lam1 The selected value for $\lambda_1$ by cross validation. Will be NULL unless method="cv".
  \item best.lam2 The selected value for $\lambda_2$ by cross validation. Will be NULL unless method="cv".
  \item cv.err Cross validation error matrix with length(lam1.vec) rows and length(lam2.vec) columns. Will be NULL unless method="cv".
\end{itemize}
Note

The algorithm is fastest when $\lambda_1$ and $\lambda_2$ are large. Use silent=FALSE to check if the algorithm is converging before the total iterations exceeds maxit.out.

Author(s)

Adam J. Rothman

References


Examples

```r
set.seed(48105)
n=50
p=10
q=5

Omega.inv=diag(q)
for(i in 1:q) for(j in 1:q)
  Omega.inv[i,j]=0.7^abs(i-j)
out=eigen(Omega.inv, symmetric=TRUE)
Omega.inv.sqrt=tcrossprod(out$vec*rep(out$val^(0.5), each=q),out$vec)
Omega=tcrossprod(out$vec*rep(out$val^(-1), each=q),out$vec)

X=matrix(rnorm(n*p), nrow=n, ncol=p)
E=matrix(rnorm(n*q), nrow=n, ncol=q)%*%Omega.inv.sqrt
Beta=matrix(rbinom(p*q, size=1, prob=0.1)*runif(p*q, min=1, max=2), nrow=p, ncol=q)
mu=1:q
Y=rep(1,n)*mu + X%*%Beta + E

lam1.vec=rev(10^seq(from=-2, to=0, by=0.5))
lam2.vec=rev(10^seq(from=-2, to=0, by=0.5))
cvfit=mrce(Y=Y, X=X, lam1.vec=lam1.vec, lam2.vec=lam2.vec, method="cv")

cvfit

fit=mrce(Y=Y, X=X, lam1=10^(-1.5), lam2=10^(-0.5), method="single")
fit

lam2.mat=1000*(fit$Bhat==0)
refit=mrce(Y=Y, X=X, lam2=lam2.mat, method="fixed.omega", omega=fit$omega, tol.in=1e-12)
refit
```
Description

Weekly log-returns of 9 stocks from 2004, analyzed in Yuan et al. (2007)

Usage

data(stock04)

Format

The format is: num [1:52, 1:9] 0.002275 -0.003795 0.012845 0.017489 -0.000369 ... - attr(*, "dimnames")=List of 2 ..$ : NULL ..$ : chr [1:9] "Walmart" "Exxon" "GM" "Ford" ...

Source


References

Index

* datasets
  stock04, 6

* package
  MRCE-package, 1

MRCE (MRCE-package), 1
mrce, 2, 2
MRCE-package, 1

stock04, 2, 6