

Package ‘glassoFast’

May 8, 2026

Type Package

Title Fast Graphical LASSO

Version 1.0.1

Date 2023-08-21

Suggests glasso, rbenchmark

Author Matyas A Sustik [aut, cph],
Ben Calderhead [aut, cph],
Julien Clavel [com, ctb]

Maintainer Julien Clavel <julien.clavel@hotmail.fr>

Description A fast and improved implementation of the graphical LASSO.

License GPL (>= 3)

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-08-21 11:50:02 UTC

Contents

glassoFast-package	1
glassoFast	2

Index	5
--------------	----------

glassoFast-package *glassoFast: a Fast Graphical LASSO*

Description

This package propose a fast implementation of the graphical LASSO of Friedman et al. 2008 based on the algorithm (FORTRAN subroutine) of Sustik and Calderhead (2012). This algorithm also avoid non-termination issues observed for the "glasso" function of the R package glasso.

Details

Package: glassoFast
Type: Package
Version: 1.0.1
Date: 2017-06-07
License: GPL (>=3.0)

The original FORTRAN Subroutine was taken from (now broken link):

<http://www.cs.utexas.edu/users/sustik/glassofast/>

Author(s)

Julien Clavel

Maintainer: Julien Clavel <julien.clavel@hotmail.fr>

References

Friedman J., Hastie T., Tibshirani R. 2008. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*. 9:432-441.

Sustik M.A., Calderhead B. 2012. GLASSOFAST: An efficient GLASSO implementation. UTCS Technical Report TR-12-29:1-3.

See Also

[glassoFast](#)

glassoFast

Fast graphical LASSO

Description

This function is a faster alternative to the "glasso" function in the glasso package (Friedman et al. 2008). This package uses the algorithm (FORTRAN subroutine) of Sustik and Calderhead (2012).

Usage

```
glassoFast(S, rho, thr = 1e-04, maxIt = 10000, start = c("cold", "warm"),  
w.init = NULL, wi.init = NULL, trace = FALSE)
```

Arguments

S	Covariance matrix (a p by p symmetric matrix)
rho	The regularization parameter for lasso. (a non-negative value or a p by p matrix of regularization parameters)
thr	Threshold for convergence. Default is 1.e-4.
maxIt	Maximum number of iterations of outer loop. Default is 10,000.
start	Type of start. Cold start is default. Using warm start, can provide starting values for w and wi.
w.init	Optional starting values for estimated covariance matrix (p by p). Only needed when start="warm" is specified
wi.init	Optional starting values for estimated inverse covariance matrix (p by p). Only needed when start="warm" is specified
trace	Flag for printing out information as iterations proceed. Default FALSE.

Details

Estimate a sparse inverse covariance matrix using a lasso (L1) penalty, following the Friedman et al. (2008) approach. The function is a wrapper of the faster and corrected (for non-termination convergence issues) FORTRAN subroutine of Sustik and Calderhead (2012).

Value

w	Estimated covariance matrix
wi	Estimated inverse covariance matrix
errflag	Memory allocation error flag: 0 means no error; !=0 means memory allocation error
niter	Number of iterations of outer loop

Author(s)

Julien Clavel

References

- Friedman J., Hastie T., Tibshirani R. 2008. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*. 9:432-441.
- Sustik M.A., Calderhead B. 2012. GLASSOFAST: An efficient GLASSO implementation. UTCS Technical Report TR-12-29:1-3.

See Also

glasso

Examples

```
set.seed(100)

# Make a random covariance matrix
p=5
x<-matrix(rnorm(p*p),ncol=p)
s<- var(x)

# Compute the LASSO estimates
glassoFast(s, rho=.1)

# compare with glasso

require(glasso)
glasso(s, rho=.1)

# benchmark glassoFast and glasso
require(rbenchmark)
p=100
x<-matrix(rnorm(p*p),ncol=p)
s<- var(x)
benchmark(glassoFast(s, rho=.15), glasso(s, rho=.15), replications = 100)
# up to an order of magnitude faster
```

Index

- * **Penalized likelihood**
 - glassoFast-package, 1
 - * **covariance matrix**
 - glassoFast, 2
 - * **glasso**
 - glassoFast, 2
 - glassoFast-package, 1
 - * **penalized likelihood**
 - glassoFast, 2
 - * **regularization**
 - glassoFast, 2
- glassoFast, 2, 2
glassoFast-package, 1