Package ‘dpglasso’

February 19, 2015

Type Package
Title Primal Graphical Lasso
Version 1.0
Date 2012-08-28
Author Rahul Mazumder and Trevor Hastie
Maintainer Rahul Mazumder <rahul.mazumder@gmail.com>
Description fits the primal graphical lasso, via one-at-a-time
   block-coordinate descent.
License GPL-2
LazyLoad yes
Repository CRAN
Date/Publication 2012-10-24 07:09:18
NeedsCompilation yes

R topics documented:

  dpglasso-package .............................................. 1
  box_qp_f .................................................. 2
  dpglasso ..................................................... 4

Index

Package dpglasso solves the primal formulation of the Graphical Lasso

Description

This package solves the primal formulation of the Graphical Lasso problem, by solving the duals
(box QPs) of the block problems.

The outer block-wise optimization routine is written in R with the inner QP solver written in Fortran
with a R wrapper around it.
Details

Package: dpglasso
Type: Package
Version: 1.0

There are two functions dpglasso and box_qp_f.

The function dpglasso solves the primal formulation of the Graphical Lasso. The function box_qp_f solves a box constrained Quadratic program via cyclical coordinate descent.

Author(s)

Rahul Mazumder and Trevor Hastie
Maintainer: Rahul Mazumder <rahul.mazumder@gmail.com>

References


Sparse inverse covariance estimation with the lasso, by J. Friedman, T. Hastie and R. Tibshirani; 2007, Biostatistics.

box_qp_f

box constrained Quadratic Program (QP)

Description

box_qp_f solves the minimization problem

\[ \text{minimize}_u \quad (b + u)'Q(b + u); \quad \text{subject to} \quad \|u\|_\infty \leq \rho \]

where \( Q_{m \times m} \) is symmetric PSD, \( u, b \in \mathbb{R}^m \). The algorithm used is one-at-a-time cyclical coordinate descent.

Usage

box_qp_f(Q, u, b, rho, Maxiter, tol = 10^-4)

Arguments

Q (Required) is a symmetric PSD matrix of dimension \( m \times m \). This is a problem parameter.

u (Required) is the optimization variable, a vector of length \( m \). The value of \( u \) serves as an initialization for the coordinate-wise algorithm. If a suitable starting point is unavailable, start with \( u = 0 \)
\textit{box\_qp\_f}

\begin{itemize}
\item \textbf{b} (Required) is a vector of length \(m\), this is a problem parameter.
\item \textbf{rho} (Required) is the degree of shrinkage. This is a non-negative scalar.
\item \textbf{Maxiter} (Required) is an integer denoting the maximum number of iterations (full sweeps across all the \(m\) variables), to be performed by \textit{box\_qp\_f}.
\item \textbf{tol} is the convergence tolerance. It is a real positive number (defaults to \(10^{-4}\)). \textit{box\_qp\_f} converges if the relative difference of the objective values is less than \(\text{tol}\).
\end{itemize}

Details

This box QP function is a \textit{R} wrapper to a Fortran code. This is primarily meant to be called from the \textit{R} function \textit{dpglasso}. One needs to be very careful (as in supplying the inputs of the program properly) while using this as a stand alone program.

Value

\begin{itemize}
\item \textbf{u} the optimal value of the argument \(u\), upon convergence
\item \textbf{grad\_vec} the gradient of the objective function at \(u\)
\end{itemize}

Author(s)

Rahul Mazumder and Trevor Hastie

References

This algorithm is used as a part of the algorithm DPGLASSO described in our paper: “The Graphical Lasso: New Insights and Alternatives by Rahul Mazumder and Trevor Hastie” available at \url{http://arxiv.org/abs/1111.5479}

Examples

```r
set.seed(2008)

# create problem data

m<-20;
aa<-array(rnorm(m*2),dim=c(m,m));
Q<-aa
Q<- Q + diag(rep(0.1,m));
b<-rnorm(m);

soln<-box_qp_f(Q, u=rep(0,m), b, rho=.2 , Maxiter=1000, tol = 10^-4)
```
Description

Does block (one row/column at a time) coordinate-wise optimization on the primal of the Graphical Lasso problem:

\[ \min_X -\log \det(X) + \text{trace}(X\Sigma) + \rho \times \|X\|_1; \text{ subject to } X \succeq 0 (A) \]

Usage

dpglasso(Sigma, X=NULL, invX=NULL, rho, outer.Maxiter=100, obj.seq=FALSE, outer.tol=10^-5)

Arguments

Sigma (Required) the sample covariance matrix, symmetric PSD with dimensions \( p \times p \).

X is an initialization to the precision matrix \( X \). It must be symmetric, PD with dimensions \( p \times p \). Defaults to \( X = \text{diag}(1/(\text{rep}(\rho, p) + \text{diag}(\text{Sigma})) ) \)

invX is an initialization to the covariance matrix. It must be symmetric with dimensions \( p \times p \). It is not necessary for \( \text{invX} \) to be the inverse of \( X \). Defaults to \( \text{invX} = \text{Sigma} + \text{diag}(\text{rep}(\rho, p)) \)

rho (Required) is the amount of regularization. It is a non-negative scalar.

outer.Maxiter the maximum number of outer iterations (i.e. row/column updates) to be performed.

\( \text{outer.Maxiter} \) defaults to 100.

obj.seq Logical variable taking values TRUE/FALSE. If \text{obj.seq}=\text{TRUE} \text{dpglasso} computes the objective value after every sweep across \( p \) rows/columns.

\text{obj.seq} \text{defaults to FALSE}

Note: Computing the objective values is \( O(p^3) \), and can take quite some time depending upon the size of the problem. Hence, it is not recommended to compute the objective values, during the course of the algorithm.

outer.tol convergence criterion. \( \text{outer.tol} \) is a non-negative scalar. If relative difference in the frobenius norm of the precision matrices across two successive iterations is below \( \text{outer.tol} \), \text{algorithm dpglasso} converges.

Details

dpglasso can also be used as a path algorithm ie solve problem (A) on a grid of \( \rho \) values. In that case, the estimates of the precision matrix \( X \) and covariance matrix \( \text{invX} \) obtained by solving (A) for a certain \( \rho \), are to be supplied as warm-starts to solve problem (A) for a smaller value of \( \rho \). See the example below.
Value

- **X**: precision matrix
- **invX**: covariance matrix
- **time.counter.QP**: This is a three dimensional vector, representing the total time taken to solve all the QPs; uses the R function `proc.time()`

Author(s)

Rahul Mazumder and Trevor Hastie

References


Examples

```r
set.seed(2008)

# create data
n=10; p = 5;
X<-array(rnorm(n*p),dim=c(n,p)); # data matrix
Sigma= cov(X); # sample covariance matrix
q<-max(abs(Sigma[rownames(Sigma)> colnames(Sigma)]));
rho=q*0.7;
B<-dpglasso(Sigma,rho=rho,outer.Maxiter=20,outer.tol=10^-6);
# uses the default initializations for the covariance and precision matrices

# now solve the problem for a smaller value of rho,
# using the previous solution as warm-start
rho.new=rho*.8;
B.new<-dpglasso(Sigma,X=X,invX=invX,
rho=rho.new,outer.Maxiter=20,outer.tol=10^-6);
```
Index

*Topic **DPGLASSO**
  dpglasso.4
*Topic **Primal Graphical Lasso**
  dpglasso.4
*Topic **box constrained QP**
  box_qp_f.2
*Topic **coordinate descent**
  box_qp_f.2
*Topic **graphical lasso**
  dpglasso.4
*Topic **package**
  dpglasso-package.1
*Topic **sparse inverse covariance selection**
  dpglasso.4

box_qp_f.2
dpglasso.4
dpglasso-package.1