Package ‘MBSP’

April 9, 2018

Type Package
Title Multivariate Bayesian Model with Shrinkage Priors
Version 1.0
Date 2018-04-04
Author Ray Bai, Malay Ghosh
Maintainer Ray Bai <raybai07@ufl.edu>
Description Implements a sparse Bayesian multivariate linear regression model using shrinkage priors from the three parameter beta normal family. The method is described in Bai and Ghosh (2018) <arXiv:1711.07635>.
License GPL-3
LazyData true
Depends R (>= 3.1.0), Matrix, MASS, GIGrvg, coda, MCMCpack
NeedsCompilation yes
Repository CRAN
Date/Publication 2018-04-09 13:17:15 UTC

R topics documented:

MBSP-package ......................................................... 2
matrix.normal ......................................................... 3
mbsp.tpbn ............................................................. 4
yeast ................................................................. 10

Index 11
Description

This package contains the function, \texttt{mbsp.tpbn}, for implementing the Multivariate Bayesian Model with Shrinkage Priors (MBSP), with the three parameter beta normal (TPBN) family, as described in Section 4.2 of Bai and Ghosh (2018).

This model gives a row sparse estimate of the $p \times q$ coefficients matrix $B$ in the multivariate regression model,

$$Y = XB + E,$$

where $Y$ is the $n \times q$ matrix of responses, $X$ is the $n \times p$ design matrix with $p$ covariates, and $E$ is the $n \times q$ noise matrix. In $E$, each row is distributed as multivariate normal $N(0, \Sigma)$, where $\Sigma$ is estimated from the data using the inverse-Wishart prior.

In addition, the package provides the function, \texttt{matrix.normal}, for sampling of a random matrix from the matrix-normal distribution.

Details

The \texttt{DESCRIPTION} file:

<table>
<thead>
<tr>
<th>Package:</th>
<th>MBSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Title:</td>
<td>Multivariate Bayesian Model with Shrinkage Priors</td>
</tr>
<tr>
<td>Version:</td>
<td>1.0</td>
</tr>
<tr>
<td>Date:</td>
<td>2018-04-04</td>
</tr>
<tr>
<td>Author:</td>
<td>Ray Bai, Malay Ghosh</td>
</tr>
<tr>
<td>Maintainer:</td>
<td>Ray Bai <a href="mailto:raybai07@ufl.edu">raybai07@ufl.edu</a></td>
</tr>
<tr>
<td>Description:</td>
<td>Implements a sparse Bayesian multivariate linear regression model using shrinkage priors from the three parameter beta normal family.</td>
</tr>
<tr>
<td>License:</td>
<td>GPL-3</td>
</tr>
<tr>
<td>LazyData:</td>
<td>true</td>
</tr>
<tr>
<td>Depends:</td>
<td>R (&gt;= 3.1.0), Matrix, MASS, GIGrvg, coda, MCMCpack</td>
</tr>
<tr>
<td>NeedsCompilation:</td>
<td>yes</td>
</tr>
<tr>
<td>Repository:</td>
<td>CRAN</td>
</tr>
</tbody>
</table>

Index of help topics:

- \texttt{MBSP-package} \hspace{1cm} Multivariate Bayesian Model with Shrinkage Priors
- \texttt{matrix.normal} \hspace{1cm} Matrix-Normal Distribution
- \texttt{mbsp.tpbn} \hspace{1cm} MBSP Model with Three Parameter Beta Normal (TPBN) Family
- \texttt{yeast} \hspace{1cm} Yeast Cell Cycle Data Set
This package implements the MBSP model for sparse Bayesian multivariate linear regression using the TPBN family. Our package performs both estimation and variable selection. The \texttt{mbsp.tpbn} function also returns the endpoints of the credible intervals (i.e. the 2.5th and 97.5th percentiles) for every single coefficient of the matrix $B$ so that uncertainty quantification can be assessed.

**Author(s)**
Ray Bai and Malay Ghosh
Maintainer: Ray Bai <raybai07@ufl.edu>

**References**


---

**matrix.normal**

**Matrix-Normal Distribution**

**Description**

This function provides a way to draw a sample from the matrix-normal distribution, given the mean matrix, the covariance structure of the rows, and the covariance structure of the columns.

**Usage**

\texttt{matrix.normal(M, U, V)}

**Arguments**

- \texttt{M} mean $a \times b$ matrix
- \texttt{U} $a \times a$ covariance matrix (covariance of rows).
- \texttt{V} $b \times b$ covariance matrix (covariance of columns).

**Details**

This function provides a way to draw a random $a \times b$ matrix from the matrix-normal distribution,

$$MN(M, U, V),$$

where $M$ is the $a \times b$ mean matrix, $U$ is an $a \times a$ covariance matrix, and $V$ is a $b \times b$ covariance matrix.
Value
A randomly drawn $a \times b$ matrix from $MN(M, U, V)$.

Author(s)
Ray Bai and Malay Ghosh

Examples

```r
# Draw a random 50x20 matrix from MN(O, U, V),
# where:
# O = zero matrix of dimension 50x20
# U has AR(1) structure,
# V has sigma^2*I structure

# Specify Mean.mat
p <- 50
q <- 20
Mean.mat <- matrix(0, nrow=p, ncol=q)

# Construct U
rho <- 0.5
times <- 1:p
H <- abs(outer(times, times, "-"))
U <- rho*H

# Construct V
sigma.sq <- 2
V <- sigma.sq*diag(q)

# Draw from MN(Mean.mat, U, V)
mm.draw <- matrix.normal(Mean.mat, U, V)
```

---

### MBSP Model with Three Parameter Beta Normal (TPBN) Family

**Description**
This function provides a fully Bayesian approach for obtaining a sparse estimate of the $p \times q$ coefficients matrix $B$ in the multivariate linear regression model,

$$ Y = X B + E, $$

using the three parameter beta normal (TPBN) family.

**Usage**

```r
mbsp.tpbn(X, Y, u=0.5, a=0.5, tau=NA, max.steps=15000, burnin=5000)
```
Arguments

\(x\)  
Design matrix of \(n\) samples and \(p\) covariates.

\(Y\)  
Response matrix of \(n\) samples and \(q\) response variables.

\(u\)  
The first parameter in the TPBN family. Defaults to \(u = 0.5\) for the horseshoe prior.

\(a\)  
The second parameter in the TPBN family. Defaults to \(a = 0.5\) for the horseshoe prior.

\(\text{tau}\)  
The global parameter. If the user does not specify this (\(\text{tau}=\text{NA}\)), the Gibbs sampler will use \(\tau = 1/(p \times n \times \log(n))\). The user may also specify a value for \(\tau\) between 0 and 1, otherwise it defaults to \(\tau = 1/(p \times n \times \log(n))\).

\(\text{max.steps}\)  
The total number of iterations to run in the Gibbs sampler. Defaults to 15,000.

\(\text{burnin}\)  
The number of burn-in iterations for the Gibbs sampler. Defaults to 5,000.

Details

The function performs sparse estimation of \(B\) and variable selection from the \(p\) covariates. The lower and upper endpoints of the 95 percent posterior credible intervals for each of the \(pq\) elements of \(B\) are also returned so that the user may assess uncertainty quantification.

In the three parameter beta normal (TPBN) family, \((u, a) = (0.5, 0.5)\) corresponds to the horseshoe prior, \((u, a) = (1, 0.5)\) corresponds to the Strawderman-Berger prior, and \((u, a) = (1, a), a > 0\) corresponds to the normal-exponential-gamma (NEG) prior. This function uses the horseshoe prior as the default shrinkage prior.

Value

The function returns a list containing the following components:

\(\text{B.est}\)  
The point estimate of the \(p \times q\) matrix \(B\) (taken as the componentwise posterior median for all \(pq\) entries).

\(\text{lower.endpoints}\)  
The 2.5th percentile of the posterior density (or the lower endpoint of the 95 percent credible interval) for all \(pq\) entries of \(B\).

\(\text{upper.endpoints}\)  
The 97.5th percentile of the posterior density (or the upper endpoint of the 95 percent credible interval) for all \(pq\) entries of \(B\).

\(\text{active.predictors}\)  
The active (nonzero) covariates chosen by our model from the \(p\) total predictors.

Author(s)

Ray Bai and Malay Ghosh
References


Examples

```r
# EXAMPLE ON SYNTHETIC DATA:  n <- 50
# Can change n, p, q, p.act, max.steps, and burnin below to reproduce the simulations from Section 5.1 of Bai and Ghosh
# Active number of predictors is 2
p.act <- 2

times <- 1:p
rho <- 0.5
H <- abs(outer(times, times, "-"))
V <- rho^H
mu <- rep(0, p)
# Rows of X are simulated from MVN(0,V)
X <- mvrnorm(n, mu, V)
# Center X
X <- scale(X, center=TRUE, scale=FALSE)

# Generate true coefficient matrix B_PN
# Entries in nonzero rows are drawn from Unif([-5,-0.5)U(0.5,5)]
B.act <- runif(p.act*q, -5, 5)
disjoint <- function(x){
  if(x <= -0.5)
```

```r
```
```r
return(x)
else
  return(x+1)
}
B.act <- matrix(sapply(B.act, disjoint), p.act, q)

# Set rest of the rows equal to 0
B.true <- rbind(B.act, matrix(0, p-p.act, q))
B.true <- B.true[sample(1:p),] # permute the rows

#########################################################
# Generate true error covariance Sigma #
#########################################################
sigma.sq=2
times <- 1:q
H <- abs(outer(times, times, "-"))
Sigma <- sigma.sq * rho^H

#########################################################
# Generate noise matrix E #
#########################################################
mu <- rep(0, q)
E <- mvrnorm(n, mu, Sigma)

#########################################################
# Generate response matrix Y #
#########################################################
Y <- crossprod(t(X), B.true) + E

#########################################################
# Run the Mbsp model on synthetic data. #
#########################################################

# For optimal estimation, change max.steps to 15,000
# and change burnin to be 5000
mbsp.model = mbsp.tpbn(X=X, Y=Y, max.steps = 2000, burnin=1000)

## Not run:
#########################################################
# Compute MSE_est and MSE_pred #
#########################################################
MSE.est <- sum((mbsp.model$B.est - B.true)^2)/(p*q)
MSE.est # Note that in the paper it is scaled by factor of 100

MSE.pred <- sum((crossprod(t(X), mbsp.model$B.est - B.true)^2)/(n*q))
MSE.pred # Note that in the paper it is scaled by a factor of 100

#########################################################
# Compute the FDR, FNR, and MP #
#########################################################

# Initialize vector for classification of predictors i = 1,...,p
```
predictor.classifications <- rep(0, p)
# Initialize vector for true classifications of predictors i = 1,...,p
true.classifications <- rep(0, p)

# True active predictors in B
true.active.indices <- which(rowSums(B.true) != 0)
# Rest true signals as 1
true.classifications[true.active.indices] <- 1

# Active predictors according to our estimates
predictor.classifications[mbsp.model$active.predictors] <- 1

# Keep track of false positives and false negatives
false.pos <- length(which(predictor.classifications != 0 & true.classifications == 0))
tot.pos <- length(which(predictor.classifications != 0))
false.neg <- length(which(predictor.classifications == 0 & true.classifications != 0))
tot.neg <- length(which(predictor.classifications == 0))

# Compute FDR, FNR, and MP
fdr.rate <- false.pos/tot.pos
fnr.rate <- false.neg/tot.neg
mis.prob <- (false.pos+false.neg)/p

fdr.rate
fnr.rate
mis.prob

## End(Not run)

## Not run:
#
#
#####################################
#####################################
# MBSP analysis of yeast cell cycle  #
# data set (Section 5.2 of Bai and Ghosh)  #
# #####################################
#

# Load yeast data set
data(yeast)

# Set seed
set.seed(12345)

# Design matrix X and response matrix Y
X <- yeast$x
X <- scale(X, center=TRUE, scale=FALSE)
Y <- yeast$y
Y <- scale(Y, center=TRUE, scale=FALSE)
# Make sure they are matrices
X <- matrix(X, nrow=nrow(X))
Y <- matrix(Y, nrow=nrow(Y))

# Run the MBSP model on the yeast cell cycle data set
mbsp.model = mbsp.tpbn(X=X, Y=Y)

# Display the active predictors (correspond to row indices in B)
mbsp.model$active.predictors

# Display names of four of the active TFs
colnames(yeast$x)[2]
colnames(yeast$x)[38]
colnames(yeast$x)[61]
colnames(yeast$x)[94]

# For horizontal axis (time)
time <- seq(from=0, to=119, by=7)

# Open pdf to plot 4 of the active TFs. This reproduces Figure 1 of Bai and Ghosh
pdf(file=file.path(tempdir(), "significantTfs.pdf"), width=5, height=5)
par(mfrow=c(2,2), mar=c(2,2,2,2))
plot(time, mbsp.model$B.est[2,], type="l", cex.axis = 0.5, xlab="", main="ACE2", ylim=c(-0.6,0.6), ylab="")
abline(h=0)
lines(time, mbsp.model$lower.endpoints[2,], lty=2)
lines(time, mbsp.model$upper.endpoints[2,], lty=2)
plot(time, mbsp.model$B.est[38,], type="l", cex.axis = 0.5, xlab="", main="HHR1", ylim=c(-0.6,0.6), ylab="")
abline(h=0)
lines(time, mbsp.model$lower.endpoints[38,], lty=2)
lines(time, mbsp.model$upper.endpoints[38,], lty=2)
plot(time, mbsp.model$B.est[61,], type="l", cex.axis = 0.5, xlab="", main="NDD1", ylim=c(-0.6,0.6), ylab="")
abline(h=0)
lines(time, mbsp.model$lower.endpoints[61,], lty=2)
lines(time, mbsp.model$upper.endpoints[61,], lty=2)
plot(time, mbsp.model$B.est[94,], type="l", cex.axis = 0.5, xlab="", main="SWI6", ylim=c(-0.6,0.6), ylab="")
abline(h=0)
lines(time, mbsp.model$lower.endpoints[94,], lty=2)
lines(time, mbsp.model$upper.endpoints[94,], lty=2)
dev.off()
**Description**

This is the Yeast Cell Cycle dataset that was originally analyzed in Chun and Keles (2010). It is also available in the package, `spls`.

**Usage**

`data(yeast)`

**Format**

A list of two components.

- **x**: ChIP-chip data. A matrix with 542 rows and 106 columns.
- **y**: Cell cycle gene expression data. A matrix with 542 rows and 18 columns.

**Details**

Matrix **y** is cell cycle gene expression data (Spellman et al., 1998) of 542 genes from an α factor based experiment. Each column corresponds to mRNA levels measured at every 7 minutes during 119 minutes (a total of 18 measurements). Matrix **x** is the chromatin immunoprecipitation on chip (ChIP-chip) data of Lee et al. (2002) and it contains the binding information for 106 transcription factors. See Chun and Keles (2010) for more details.

**Source**


**References**

Index

matrix.normal, 3
MBSP (MBSP-package), 2
MBSP-package, 2
mbsp.tpbn, 4

yeast, 10