Package ‘MBSP’

August 11, 2022

Type Package

Title Multivariate Bayesian Model with Shrinkage Priors

Version 3.0

Date 2022-08-11

Author Ray Bai, Malay Ghosh

Maintainer Ray Bai <raybaistat@gmail.com>

Description Gibbs sampler for fitting multivariate Bayesian linear regression with shrinkage priors (MBSP), using the three parameter beta normal family. The method is described in Bai and Ghosh (2018) <doi:10.1016/j.jmva.2018.04.010>.

License GPL-3

Depends R (>= 3.6.0)

Imports stats, MCMCpack, GIGrvg, utils, MASS

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-08-11 12:40:02 UTC

R topics documented:

`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

```
matrix_normal(M, U, V)
```

Arguments

M	mean $a \times b$ matrix
U	$a \times a$ covariance matrix (covariance of rows).
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

```
# 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)
mn_draw <- matrix_normal(Mean_mat, U, V)
```

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

Description

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

$$Y = XB + E,$$

using the three parameter beta normal (TPBN) family. Here Y is the $n \times q$ matrix with n samples of q response variables, X is the $n \times p$ design matrix with n samples of p covariates, and E is the $n \times q$ noise matrix with independent rows. The complete model is described in Bai and Ghosh (2018).

If there are r confounding variables which *must* remain in the model and should *not* be regularized, then these can be included in the model by putting them in a separate $n \times r$ confounding matrix Z. Then the model that is fit is

$$Y = XB + ZC + E,$$

where C is the $r \times q$ regression coefficients matrix corresponding to the confounders. In this case, we put a flat prior on C. By default, confounders are not included.

Usage

```
MBSP(Y, X, confounders=NULL, u=0.5, a=0.5, tau=NA,
     max_steps=6000, burnin=1000, save_samples=TRUE)
```

Arguments

Y	Response matrix of n samples and q response variables.
X	Design matrix of n samples and p covariates. The MBSP model regularizes the regression coefficients B corresponding to X.
confounders	Optional design matrix Z of n samples of r confounding variables. By default, confounders are not included in the model (`confounders=NULL`). However, if there are some confounders that *must* remain in the model and should *not* be regularized, then the user can include them here.
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.
tau	The global parameter. If the user does not specify this (tau=NA), the Gibbs sampler will use $\tau = 1/(p * n * log(n))$. The user may also specify a value for τ between 0 and 1, otherwise it defaults to $1/(p * n * log(n))$.

`max_steps`	The total number of iterations to run in the Gibbs sampler. Defaults to `6000`.
`burnin`	The number of burn-in iterations for the Gibbs sampler. Defaults to `1000`.
`save_samples`	A Boolean variable for whether to save all of the posterior samples of the regression coefficients matrix B. Defaults to `"TRUE"`.

Details

The function performs (nearly) sparse estimation of the regression coefficients matrix 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 shinkrage prior.

The user also has the option of including an $n \times r$ matrix with r confounding variables. These confounders are variables which are included in the model but should *not* be regularized.

Value

The function returns a list containing the following components:

`B_est`	The point estimate of the $p \times q$ matrix B (taken as the componentwise posterior median for all pq entries).
`B_CI_lower`	The 2.5th percentile of the posterior density (or the lower endpoint of the 95 percent credible interval) for all pq entries of B.
`B_CI_upper`	The 97.5th percentile of the posterior density (or the upper endpoint of the 95 percent credible interval) for all pq entries of B.
`active_predictors`	The row indices of the active (nonzero) covariates chosen by our model from the p total predictors.
`B_samples`	All `max_steps-burnin` samples of B.
`C_est`	The point estimate of the $r \times q$ matrix C corresponding to the confounders (taken as the componentwise posterior median for all rq entries. This matrix is not returned if there are no confounders (i.e. `confounders=NULL`).
`C_CI_lower`	The 2.5th percentile of the posterior density (or the lower endpoint of the 95 percent credible interval) for all rq entries of C. This is not returned if there are no confounders (i.e. `confounders=NULL`).
`C_CI_upper`	The 97.5th percentile of the posterior density (or the upper endpoint of the 95 percent credible interval) for all rq entries of C. This is not returned if there are no confounders (i.e. `confounders=NULL`)
`C_samples`	All `max_steps-burnin` samples of C. This is not returned if there are no confounders (i.e. `confounders=NULL`)

Author(s)

Ray Bai and Malay Ghosh

References

Armagan, A., Clyde, M., and Dunson, D.B. (2011) Generalized Beta Mixtures of Gaussians. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger (Eds.) *Advances in Neural Information Processing Systems 24*, 523-531.

Bai, R. and Ghosh, M. (2018). High-Dimensional Multivariate Posterior Consistency Under Global-Local Shrinkage Priors. *Journal of Multivariate Analysis*, **167**: 157-170. Berger, J. (1980). A Robust Generalized Bayes Estimator and Confidence Region for a Multivariate Normal Mean. *Annals of Statistics*, **8**(4): 716-761.

Carvalho, C.M., Polson, N.G., and Scott., J.G. (2010). The Horseshoe Estimator for Sparse Signals. *Biometrika*, **97**(2): 465-480.

Strawderman, W.E. (1971). Proper Bayes Minimax Estimators of the Multivariate Normal Mean. *Annals of Mathematical Statistics*, **42**(1): 385-388.

Examples

```
n <- 100
p <- 40
q <- 3      # number of response variables is 3
p_act <- 5  # number of active (nonzero) predictors is 5

#############################
# Generate design matrix X. #
#############################
set.seed(123)
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 <- MASS::mvrnorm(n, mu, V)
# Center X
X <- scale(X, center=TRUE, scale=FALSE)

############################################
# Generate true coefficient matrix B_true. #
############################################
# Entries in nonzero rows are drawn from Unif[(-5,-0.5)U(0.5,5)]
B_act <- runif(p_act*q,-5,4)
disjoint <- function(x){
         if(x <= -0.5)
            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 <- MASS::mvrnorm(n, mu, Sigma)

##############################
# Generate response matrix Y #
##############################
Y <- crossprod(t(X),B_true) + E

# Note that there are no confounding variables in this synthetic example


#########################################
# Fit the MBSP model on synthetic data. #
#########################################

# Should use default of max_steps=6000, burnin=1000 in practice
mbsp_model = MBSP(Y=Y, X=X, max_steps=1000, burnin=500)

# indices of the true nonzero rows
true_active_predictors <- which(rowSums(B_true)!=0)
true_active_predictors

# variables selected by the MBSP model
mbsp_model$active_predictors

# the true nonzero rows
B_true[true_active_predictors, ]

# the MBSP model's estimates of the nonzero rows
mbsp_model$B_est[true_active_predictors, ]
```

Index