

# Package ‘NormalBetaPrime’

August 15, 2018

**Type** Package

**Title** Normal Beta Prime Prior

**Version** 1.1

**Date** 2018-08-15

**Author** Ray Bai, Malay Ghosh

**Maintainer** Ray Bai <raybai07@ufl.edu>

**Description** Implements Bayesian linear regression, variable selection, normal means estimation, and multiple hypothesis testing using the normal-beta prime prior, as introduced by Bai and Ghosh (2018) <arXiv:1807.02421> and Bai and Ghosh (2018) <arXiv:1807.06539>. Normal means estimation and multiple testing for the Dirichlet-Laplace <doi:10.1080/01621459.2014.960967> and horseshoe+ priors <doi:10.1214/16-BA1028> are also available in this package.

**License** GPL-3

**LazyData** true

**Depends** R (>= 3.1.0)

**Imports** stats, utils, Matrix, MASS, glmnet, pscl, GIGrv, pracma

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2018-08-15 17:50:02 UTC

## R topics documented:

NormalBetaPrime-package . . . . .	2
diabetes . . . . .	3
dl.normalmeans . . . . .	4
eyedata . . . . .	7
genes . . . . .	8
hsplus.normalmeans . . . . .	9
nbp . . . . .	12
nbp.normalmeans . . . . .	16
singh2002 . . . . .	20
trim32 . . . . .	21

---

NormalBetaPrime-package

*Normal-Beta Prime*

---

## Description

This package contains the `nbp` and `nbp.normalmeans` functions for implementing Bayesian linear regression, variable selection, sparse normal means estimation, and multiple hypothesis testing with the normal-beta prior (NBP) prior, as introduced by Bai and Ghosh (2018) <arXiv:1807.02421> and Bai and Ghosh (2018) <arXiv:1807.06539>. Functions for normal means estimation and hypothesis testing are also available for the Dirichlet-Laplace (DL) prior of Bhattacharya et al. (2015) and the horseshoe+ (HS+) prior of Bhadra et al. (2017): `dl.normalmeans` and `hsplus.normalmeans` respectively.

## Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

This package implements the NBP model for Bayesian linear regression, variable selection, sparse normal means estimation, and multiple hypothesis testing. Our package also contains functions for sparse normal means estimation and multiple hypothesis testing for the DL and HS+ priors.

## Author(s)

Ray Bai and Malay Ghosh

Maintainer: Ray Bai <raybai07@ufl.edu>

## References

- Bai, R. and Ghosh, M. (2018). "Large-Scale Multiple Hypothesis Testing With the Normal Beta-Prime Prior." Pre-print, arXiv:1807.02421.
- Bai, R. and Ghosh, M. (2018). "On the Beta Prime Prior for Scale Parameters in High-Dimensional Regression Models." Pre-print, arXiv:1807.06539.
- Bhadra, A., Datta, J., Polson, N. G., and Willard, B. (2017). "The Horseshoe+ Estimator for Ultra-Sparse Signals." *Bayesian Analysis*, **12**(4):1105-1131.
- Bhattacharya, A., Pati, D., Pillai, N. S., and Dunson, D. B. (2015). "Dirichlet-Laplace Priors for Optimal Shrinkage." *Journal of the American Statistical Association*, **110**(512):1479-1490.
- van der Pas, S., Scott, J., Chakraborty, A., and Bhattacharya, A. horseshoe: Implementation of the Horseshoe Prior. <https://CRAN.R-project.org/package=horseshoe>.
- van der Pas, S. L., Kleijn, B. J. K., and van der Vaart, A. W. (2014). "The Horseshoe Estimator: Posterior Concentration Around Nearly Black Vectors." *Electronic Journal of Statistics*, **8**(2):2585-2618.

van der Pas, S. L., Szabo, B. T., and van der Vaart, A. (2017). "Adaptive posterior contraction rates for the horseshoe." *Electronic Journal of Statistics*, **11**(2):3196-3225.

---

diabetes

*Blood and other measurements in diabetics*

---

### Description

The diabetes data frame has 442 rows and 3 columns. These are the data used in the Efron et al. "Least Angle Regression" paper and is also available in the lars package.

### Usage

```
data(diabetes)
```

### Format

This data frame consists of the following columns:

**x:** is a design matrix with 10 columns (no interactions).

**y:** is a numeric vector.

**x2:** is a design matrix with 64 columns (includes interactions).

### Details

The x matrix has been standardized to have unit L2 norm in each column and zero mean. The matrix x2 consists of x plus certain interactions.

### Source

<https://cran.r-project.org/web/packages/lars/>

### References

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2003). "Least Angle Regression" (with discussion). *Annals of Statistics*, **32**(2):407-499.

---

dl.normalmeans	<i>Normal Means Estimation and Hypothesis Testing with the Dirichlet-Laplace Prior</i>
----------------	--

---

### Description

This function implements the Dirichlet-Laplace model of Bhattacharya et al. (2015) for obtaining a sparse estimate of  $\theta = (\theta_1, \dots, \theta_n)$  in the normal means problem,

$$X_i = \theta_i + \epsilon_i,$$

where  $\epsilon_i \sim N(0, \sigma^2)$ . This is achieved by placing the Dirichlet-Laplace (DL) prior on the individual  $\theta_i$ 's. The sparsity parameter  $a$  in the  $Dir(a, \dots, a)$  prior can be specified *a priori*, or it can be estimated from the data, either by: 1) using the estimate of sparsity level by van der Pas et al. (2014), or 2) by taking a restricted marginal maximum likelihood (REML) estimate. Multiple testing can also be performed by either thresholding the shrinkage factor in the posterior mean, or by examining the marginal 95 percent credible intervals.

### Usage

```
dl.normalmeans(x, a.est=c("fixed", "est.sparsity", "reml"), a=1/length(x),
               sigma2=1, var.select = c("threshold", "intervals"),
               max.steps=10000, burnin=5000)
```

### Arguments

x	an $n \times 1$ multivariate normal vector.
a.est	The method for estimating the sparsity parameter $a$ . If "fixed" is chosen, then $a$ is not estimated from the data. If "est.sparsity" is chosen, the empirical Bayes estimate of sparsity level from van der Pas et al. (2014) is used. If "reml" is chosen, $a$ is estimated from restricted marginal maximum likelihood.
a	The concentration parameter $a$ in the $Dir(a, \dots, a)$ prior. Controls the sparsity of the model. Defaults to $1/n$ , but user may specify a different value for $a$ ( $a > 0$ ). This is ignored if the method "est.sparsity" or "reml" is used to estimate $a$ .
sigma2	The variance parameter. Defaults to 1. User needs to specify the noise parameter if it is different from 1 ( $\sigma^2 > 0$ ).
var.select	The method of variable selection. "threshold" selects variables by thresholding the shrinkage factor in the posterior mean. "intervals" will classify $\theta_i$ 's as either signals ( $\theta_i \neq 0$ ) or as noise ( $\theta_i = 0$ ) by examining the 95 percent posterior credible intervals.
max.steps	The total number of iterations to run in the Gibbs sampler. Defaults to 10,000.
burnin	The number of burn-in iterations for the Gibbs sampler. Defaults to 5,000.

## Details

The function implements sparse estimation and multiple hypothesis testing on a multivariate normal mean vector,  $\theta = (\theta_1, \dots, \theta_n)$  with the Dirichlet-Laplace prior of Bhattacharya et al. (2015). The full model is:

$$\begin{aligned} X|\theta &\sim N_n(\theta, \sigma^2 I_n), \\ \theta_i | (\psi_i, \phi_i, \tau) &\sim N(0, \sigma^2 \psi_i \phi_i^2 \tau^2), i = 1, \dots, n, \\ \psi_i &\sim \text{Exp}(1/2), i = 1, \dots, n, \\ (\phi_1, \dots, \phi_n) &\sim \text{Dir}(a, \dots, a), \\ \tau &\sim G(na, 1/2). \end{aligned}$$

$a$  is the main parameter that controls the sparsity of the solution. It can be estimated by the empirical Bayes estimate of the estimated sparsity ("est.sparsity") given in van der Pas et al. (2014), or it can be estimated by restricted marginal maximum likelihood ("reml") in the interval  $[1/n, 1]$ .

The posterior mean is of the form  $[E(1 - \kappa_i | X_i)] X_i, i = 1, \dots, n$ . The "threshold" method for variable selection is to classify  $\theta_i$  as signal ( $\theta_i \neq 0$ ) if  $E(1 - \kappa_i | X_i) > 1/2$ .

## Value

The function returns a list containing the following components:

theta.hat	The posterior mean estimate of $\theta$ .
theta.med	The posterior median estimate of $\theta$ .
theta.intervals	The 95 percent credible intervals for all $n$ components of $\theta$ .
dl.classifications	An $n$ -dimensional binary vector with "1" if the covariate is selected and "0" if it is deemed irrelevant.
a.estimate	The estimate of the sparsity level. If user specified "fixed" for a.est, then it simply returns the fixed $a$ .

## Author(s)

Ray Bai and Malay Ghosh

## References

- Bhattacharya, A., Pati, D., Pillai, N. S., and Dunson, D. B. (2015). "Dirichlet-Laplace Priors for Optimal Shrinkage." *Journal of the American Statistical Association*, **110**(512):1479-1490.
- van der Pas, S. L., Kleijn, B. J. K., and van der Vaart, A. W. (2014). "The Horseshoe Estimator: Posterior Concentration Around Nearly Black Vectors." *Electronic Journal of Statistics*, **8**(2):2585-2618.
- van der Pas, S. L., Szabo, B. T., and van der Vaart, A. (2017). "Adaptive posterior contraction rates for the horseshoe." *Electronic Journal of Statistics*, **11**(2):3196-3225.

## Examples

```
#####
#####
## Example on synthetic data. ##
## 5 percent of entries in a sparse vector theta ##
## are set equal to signal value A =7. ##
#####
#####
n <- 100
sparsity.level <- 5
A <- 7

# Initialize theta vector of all zeros
theta.true <- rep(0,n)
# Set (sparsity.level) percent of them to be A
q <- floor(n*(sparsity.level/100))
# Pick random indices of theta.true to equal A
signal.indices <- sample(1:n, size=q, replace=FALSE)

#####
# Generate data X #
#####
theta.true[signal.indices] <- A
X <- theta.true + rnorm(n,0,1)

#####
# Run the DL model on X #
#####
# For optimal performance, should set max.steps=10,000 and burnin=5000.

# Estimate b from the empirical Bayes estimate of sparsity level
dl.model <- dl.normalmeans(X, a.est="est.sparsity", sigma2=1, var.select="threshold",
                           max.steps=1000, burnin=500)

dl.model$theta.med          # Posterior median estimates
dl.model$dl.classifications # Classifications
dl.model$a.estimate         # Estimate of sparsity level

#
#
#####
#####
## Prostate cancer data analysis. ##
## Running this code will allow you to reproduce ##
## the results in Section 6 of Bai and Ghosh (2018) ##
#####
#####

# Load the data
data(singh2002)
attach(singh2002)
```

```

# First 50 rows are the cancer patients,
# and the last 52 rows are the control subjects.

prostate.data <- singh2002$x

# Perform 2-sample t-test and obtain z-scores
n <- ncol(prostate.data)
test.stat <- rep(NA,n)
z.scores <- rep(NA, n)

# Fill in the vectors
for(i in 1:n){
  test.stat[i] <- t.test(prostate.data[51:102,i],
                        prostate.data[1:50,i])$statistic
  z.scores[i] <- qnorm(pt(test.stat[i],100))
}

#####
# Apply DL model on the z-scores.      #
# Here sigma2 is known with sigma2= 1 #
#####

dl.model <- dl.normalmeans(z.scores, a.est="reml", var.select="threshold")

#####
# How many genes flagged as significant? #
#####
num.sig <- sum(dl.model$dl.classifications != 0)
num.sig

#####
# Estimated effect size for 10 most significant genes #
#####
most.sig <- c(610,1720,332,364,914,3940,4546,1068,579,4331)
dl.model$theta.hat[most.sig]

```

---

eyedata

*The Bardet-Biedl syndrome gene expression data from Scheetz et al. (2006)*

---

### Description

Gene expression data from the microarray study by Scheetz et al. (2006). This data is also available in the flare package.

### Usage

data(eyedata)

**Format**

This data is a list that consists of the following:

`genes`: is a design matrix with 120 rows and 200 columns.

`trim32`: is a numeric vector with 120 samples of gene expression levels of the TRIM32 gene.

**Details**

The data set contains gene expression data (200 genes for 120 samples) and gene expression levels of TRIM32 from the microarray experiments of mammalian eye tissue samples of Scheetz et al. (2006).

**Source**

<https://cran.r-project.org/web/packages/flare/>

**References**

Scheetz, T. E., Kim K.-Y. A., Swiderski R. E., Philp A. R., Braun, T. A., Knudtson, K. L., Dorrance, A. M., DiBona, G. F., Huang, J., Casavant, T. L., Sheffield, V. C., and Stone, E. M. (2006). "Regulation of gene expression in the mammalian eye and its relevance to eye disease." *Proceedings of the National Academy of Sciences*, **103**(39):14429-14434.

---

genes

*Gene expression data for predicting Bardet-Biedl syndrome*

---

**Description**

Gene expression data from the microarray study by Scheetz et al. (2006). This data is also available in the flare package.

**Usage**

`genes`

**Format**

`genes`: is a design matrix with 120 rows and 200 columns.

**Details**

The data set contains gene expression data (200 genes for 120 samples) from the microarray experiments of mammalian eye tissue samples of Scheetz et al. (2006).

**Source**

<https://cran.r-project.org/web/packages/flare/>



## References

Scheetz, T. E., Kim K.-Y. A., Swiderski R. E., Philp A. R., Braun, T. A., Knudtson, K. L., Dorrance, A. M., DiBona, G. F., Huang, J., Casavant, T. L., Sheffield, V. C., and Stone, E. M. (2006). "Regulation of gene expression in the mammalian eye and its relevance to eye disease." *Proceedings of the National Academy of Sciences*, **103**(39):14429-14434.

---

hsplus.normalmeans      *Normal Means Estimation and Hypothesis Testing with the Horseshoe+ Prior*

---

## Description

This function implements the horseshoe+ model of Bhadra et al. (2017) for obtaining a sparse estimate of  $\theta = (\theta_1, \dots, \theta_n)$  in the normal means problem,

$$X_i = \theta_i + \epsilon_i,$$

where  $\epsilon_i \sim N(0, \sigma^2)$ . This is achieved by placing the horseshoe+ (HS+) prior on the individual  $\theta_i$ 's. The sparsity parameter  $\tau$  can be specified *a priori*, or it can be estimated from the data, either by: 1) using the estimate of sparsity level by van der Pas et al. (2014), or 2) by taking a restricted marginal maximum likelihood (REML) estimate. Multiple testing can also be performed by either thresholding the shrinkage factor in the posterior mean, or by examining the marginal 95 percent credible intervals.

## Usage

```
hsplus.normalmeans(x, tau.est=c("fixed", "est.sparsity", "reml"), tau=1,
                  sigma2=1, var.select=c("threshold", "intervals"),
                  max.steps = 10000, burnin=5000)
```

## Arguments

<code>x</code>	an $n \times 1$ multivariate normal vector.
<code>tau.est</code>	The method for estimating the sparsity parameter $\tau$ . If "fixed" is chosen, then $\tau$ is not estimated from the data. If "est.sparsity" is chosen, the empirical Bayes estimate of sparsity level from van der Pas et al. (2014) is used. If "reml" is chosen, $\tau$ is estimated from restricted marginal maximum likelihood.
<code>tau</code>	The global parameter $\tau$ in the HS+ prior. Controls the sparsity of the model. Defaults to 1. User may specify a different value for tau ( $\tau > 0$ ). This is ignored if the method "est.sparsity" or "reml" is used to estimate $\tau$ .
<code>sigma2</code>	The variance parameter. Defaults to 1. User needs to specify the noise parameter if it is different from 1 ( $\sigma^2 > 0$ ).
<code>var.select</code>	The method of variable selection. "threshold" selects variables by thresholding the shrinkage factor in the posterior mean. "intervals" will classify $\theta_i$ 's as either signals ( $\theta_i \neq 0$ ) or as noise ( $\theta_i = 0$ ) by examining the 95 percent posterior credible intervals.

max.steps	The total number of iterations to run in the Gibbs sampler. Defaults to 10,000.
burnin	The number of burn-in iterations for the Gibbs sampler. Defaults to 5,000.

### Details

The function implements sparse estimation and multiple hypothesis testing on a multivariate normal mean vector,  $\theta = (\theta_1, \dots, \theta_n)$  with the horseshoe+ prior of Bhadra et al. (2015). The full model is:

$$\begin{aligned} X|\theta &\sim N_n(\theta, \sigma^2 I_n), \\ \theta_i|\lambda_i &\sim N(0, \sigma^2 \lambda_i^2), i = 1, \dots, n, \\ \lambda_i &\sim C^+(0, \tau \eta_i), i = 1, \dots, n, \\ \eta_i &\sim C^+(0, 1), i = 1, \dots, n. \end{aligned}$$

$\tau$  is the global parameter that controls the sparsity of the solution. It can be estimated by the empirical Bayes estimate of the estimated sparsity ("est.sparsity") given in van der Pas et al. (2014), or it can be estimated by restricted marginal maximum likelihood ("reml") in the interval  $[1/n, 1]$ .

The posterior mean is of the form  $[E(1 - \kappa_i|X_i)]X_i, i = 1, \dots, n$ . The "threshold" method for variable selection is to classify  $\theta_i$  as signal ( $\theta_i \neq 0$ ) if  $E(1 - \kappa_i|X_i) > 1/2$ .

### Value

The function returns a list containing the following components:

theta.hat	The posterior mean estimate of $\theta$ .
theta.med	The posterior median estimate of $\theta$ .
theta.intervals	The 95 percent credible intervals for all $n$ components of $\theta$ .
hsplus.classifications	An $n$ -dimensional binary vector with "1" if the covariate is selected and "0" if it is deemed irrelevant.
tau.estimate	The estimate of the sparsity level. If user specified "fixed" for tau.est, then it simply returns the fixed $\tau$ .

### Author(s)

Ray Bai and Malay Ghosh

### References

- Bhadra, A., Datta, J., Polson, N. G., and Willard, B. (2017). "The Horseshoe+ Estimator for Ultra-Sparse Signals." *Bayesian Analysis*, **12**(4):1105-1131.
- Makalic, E., Schmidt, D. F., and Hopper, J. L. (2016). "Bayesian Robust Regression with the Horseshoe+ Estimator." *AI 2016: Advances in Artificial Intelligence*, 429-440.

van der Pas, S. L., Kleijn, B. J. K., and van der Vaart, A. W. (2014). "The Horseshoe Estimator: Posterior Concentration Around Nearly Black Vectors." *Electronic Journal of Statistics*, **8**(2):2585-2618.

van der Pas, S. L., Szabo, B. T., and van der Vaart, A. (2017). "Adaptive posterior contraction rates for the horseshoe." *Electronic Journal of Statistics*, **11**(2):3196-3225.

## Examples

```
#####
## Example on synthetic data. ##
## 5 percent of entries in a sparse vector theta ##
## are set equal to signal value A =7. ##
#####
n <- 100
sparsity.level <- 5
A <- 7

# Initialize theta vector of all zeros
theta.true <- rep(0,n)
# Set (sparsity.level) percent of them to be A
q <- floor(n*(sparsity.level/100))
# Pick random indices of theta.true to equal A
signal.indices <- sample(1:n, size=q, replace=FALSE)

#####
# Generate data X #
#####
theta.true[signal.indices] <- A
X <- theta.true + rnorm(n,0,1)

#####
# Run the horseshoe+ model on X #
#####
# For optimal performance, should set max.steps=10,000 and burnin=5000.

# Fix tau at 1/n
hsplus.model <- hsplus.normalmeans(X, tau.est="fixed", tau=1/length(X),
                                   sigma2=1, var.select="threshold",
                                   max.steps=1000, burnin=500)

hsplus.model$theta.intervals      # 95 percent credible intervals
hsplus.model$hspplus.classifications # Classifications
hsplus.model$tau.estimate         # Estimate of sparsity level

#####
#####
## Prostate cancer data analysis. ##
## Running this code will allow you to reproduce ##
## the results in Section 6 of Bai and Ghosh (2018) ##
#####
#####
```

```

# Load the data
data(singh2002)
attach(singh2002)

# First 50 rows are the cancer patients,
# and the last 52 rows are the control subjects.

prostate.data <- singh2002$x

# Perform 2-sample t-test and obtain z-scores
n <- ncol(prostate.data)
test.stat <- rep(NA,n)
z.scores <- rep(NA, n)

# Fill in the vectors
for(i in 1:n){
  test.stat[i] <- t.test(prostate.data[51:102,i],
                        prostate.data[1:50,i])$statistic
  z.scores[i] <- qnorm(pt(test.stat[i],100))
}

#####
# Apply horseshoe+ model on the z-scores. #
# Here sigma2 is known with sigma2= 1     #
#####

hsplus.model <- hsplus.normalmeans(z.scores, tau.est="reml", var.select="threshold")

#####
# How many genes flagged as significant? #
#####
num.sig <- sum(hsplus.model$hsplus.classifications != 0)
num.sig

#####
# Estimated effect size for 10 most significant genes #
#####
most.sig <- c(610,1720,332,364,914,3940,4546,1068,579,4331)
hsplus.model$theta.hat[most.sig]

```

**Description**

This function implements the normal-beta prime (NBP) model for estimation and variable selection in the standard linear regression model,

$$y = X\beta + \epsilon,$$

where  $\epsilon \sim N_n(0, \sigma^2 I_n)$ . This is achieved by placing the normal-beta prime (NBP) prior of Bai and Ghosh (2018) on the coefficients of  $\beta$ . In the case where  $p > n$ , we utilize an efficient sampler from Bhattacharya et al. (2016) to reduce the computational cost of sampling from the full conditional density of  $\beta$  to  $O(n^2 p)$ . The hyperparameters can be set deterministically by the user or they may be automatically selected by marginal maximum likelihood (MML).

### Usage

```
nbp(X, y, method.hyperparameters=c("mml", "fixed"), a=0.5, b=0.5,
    c=1e-5, d=1e-5, max.steps = 20000, burnin=10000,
    selection=c("dss", "intervals"))
```

### Arguments

X	$n \times p$ design matrix. Should be centered.
y	$n \times 1$ response vector. Should be centered.
method.hyperparameters	The method for estimating the shape parameters $(a, b)$ . If "mml" is chosen, the function estimates the marginal maximum likelihood (MML) estimates of $(a, b)$ by the EM/Gibbs algorithm described in Bai and Ghosh (2012). If "fixed" is chosen, then $(a, b)$ are not estimated from the data.
a	Shape parameter for $\beta'(a, b)$ . The default is 0.5. The user may specify a different value for a ( $a > 0$ ). This is ignored if the method for estimating hyperparameters is "mml".
b	Shape parameter for $\beta'(a, b)$ . The default is 0.5. The user may specify a different value for b ( $b > 0$ ). This is ignored if the method for estimating hyperparameters is "mml".
c	The shape parameter for the $IG(c, d)$ prior on unknown variance parameter, $\sigma^2$ . The default is $10^{-5}$ .
d	The rate parameter for the $IG(c, d)$ prior on the unknown variance parameter, $\sigma^2$ . The default is $10^{-5}$ .
max.steps	The total number of iterations to run in the Gibbs sampler. Defaults to 20,000.
burnin	The number of burn-in iterations for the Gibbs sampler. Defaults to 10,000.
selection	The method of variable selection. "dss" implements the decoupled selection and shrinkage (DSS) method of Hahn and Carvalho (2015) to select variables. "intervals" performs variable selection by examining the 95 percent posterior credible intervals for each coefficient, $\beta_i, i = 1, \dots, p$ .

### Details

The function performs estimation and variable selection in the standard linear regression model using the normal-beta prime (NBP) model of Bai and Ghosh (2018). The 95 percent credible intervals

for each of the  $p$  covariates are also returned so that the user may assess uncertainty quantification. The full model is:

$$\begin{aligned} Y|(X, \beta) &\sim N_n(X\beta, \sigma^2 I_n), \\ \beta_i|\omega_i^2 &\sim N(0, \sigma^2 \omega_i^2), i = 1, \dots, p, \\ \omega_i^2 &\sim \beta'(a, b), i = 1, \dots, p, \\ \sigma^2 &\sim IG(c, d), \end{aligned}$$

where  $\beta'(a, b)$  denotes the beta prime density,

$$\pi(\omega_i^2) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} (\omega_i^2)^{a-1} (1 + \omega_i^2)^{-a-b}.$$

## Value

The function returns a list containing the following components:

beta.hat	The posterior mean estimate of $\beta$ .
beta.med	The posterior median estimate of $\beta$ .
beta.intervals	The 95 percent credible intervals for all $p$ estimates in $\beta$ .
nbp.classifications	A $p$ -dimensional binary vector with "1" if the covariate is selected and "0" if it is deemed irrelevant.
sigma2.hat	Estimate of unknown variance component $\sigma^2$ .
a.estimate	MML estimate of shape parameter $a$ . If $a$ was fixed <i>a priori</i> , returns fixed $a$ .
b.estimate	MML estimate of shape parameter $b$ . If $b$ was fixed <i>a priori</i> , returns fixed $b$ .

## Author(s)

Ray Bai and Malay Ghosh

## References

- Bai, R. and Ghosh, M. (2018). "On the Beta Prime Prior for Scale Parameters in High-Dimensional Bayesian Regression Models." Pre-print, arXiv:1807.06539.
- Bhattacharya, A., Chakraborty, A., and Mallick, B.K. (2016). "Fast Sampling with Gaussian Scale Mixture Priors in High-Dimensional Regression." *Biometrika*, **69**(2): 447-457.
- Hahn, P. R. and Carvalho, C. M. (2015). "Decoupling shrinkage and selection in Bayesian linear models: A posterior Summary Perspective." *Journal of the American Statistical Association*, **110**(509):435-448.

## Examples

```
#####
## Example on diabetes data. ##
#####
data(diabetes)
attach(diabetes)
X <- scale(diabetes$x) # Center and scale X
y <- scale(diabetes$y) # Center and scale y

#####
# Fit the NBP regression model #
#####
# Should use default of 20,000 for max.steps and 10,000 for burnin
nbp.model <- nbp(X=X, y=y, method.hyperparameters="mml",
                 max.steps=5000, burnin=2500, selection="dss")

nbp.model$beta.med      # posterior median estimates
nbp.model$a.estimate    # MML estimate of shape parameter 'a'
nbp.model$b.estimate    # MML estimate of shape parameter 'b'

nbp.model$beta.intervals # 95 percent posterior credible intervals
nbp.model$nbp.classifications # Variables selected

#
#
#####
#####
## TRIM32 gene expression data analysis. ##
## Running this code will allow you to reproduce ##
## the results in Section 6 of Bai and Ghosh (2018) ##
#####
#####

# Load the data
data(eyedata)

# Set seed
set.seed(123)

# Center design matrix X and response vector y
X <- scale(genes, center=TRUE, scale=TRUE) # gene expression data (covariates)
y <- scale(trim32, center=TRUE, scale=TRUE) # levels of TRIM32 (response)

#####
# Implement the NBP model #
#####
nbp.model = nbp(X,y, method.hyperparameters="mml", selection="dss")

# Variables selected
active.indices <- which(nbp.model$nbp.classifications != 0) # active genes
```

```

active.estimates <- nbp.model$beta.med[active.indices]
active.CIs <- nbp.model$beta.intervals[, active.indices]

#####
# Evaluate predictive performance #
#####
k <- 5 # Number of folds

# Randomly select indices for the 5 different folds
folds <- split(sample(nrow(X), nrow(X), replace=FALSE), as.factor(1:k))
# To store the mean square prediction error
mspe.nbp <- rep(NA, k)

for(i in 1:k){

  # Split data into training set and test set based on folds
  test_ind = folds[[i]]

  # 80 percent training set
  y.train = y[-test_ind]
  X.train = X[-test_ind, ]

  # Rest is test set
  y.test = y[test_ind]
  X.test = X[test_ind, ]

  # Run MBSP model on training set
  nbp.train = nbp(X=X.train, y=y.train, method.hyperparameters="mml", selection="dss")
  beta.train <- nbp.train$beta.med

  # Obtain predictions on test set
  nbp.pred = crossprod(t(X.test), beta.train)

  # Compute the MSPE on test set
  mspe.nbp[i] = mean((nbp.pred - y.test)^2)
}

mean.nbp.mspe <- mean(mspe.nbp)
mean.nbp.mspe

```

---

nbp.normalmeans

*Normal Means Estimation and Hypothesis Testing with the NBP Prior*


---

### Description

This function implements the NBP model of Bai and Ghosh (2018) for obtaining a sparse estimate of  $\theta = (\theta_1, \dots, \theta_n)$  in the normal means problem,



$$X_i = \theta_i + \epsilon_i,$$

where  $\epsilon_i \sim N(0, \sigma^2)$ . This is achieved by placing the normal-beta prime (NBP) prior on the individual  $\theta_i$ 's. The sparsity parameter  $b$  can be specified *a priori*, or it can be estimated from the data, either by: 1) using the estimate of sparsity level by van der Pas et al. (2014), or 2) by taking a restricted marginal maximum likelihood (REML) estimate. Multiple testing can also be performed by either thresholding the shrinkage factor in the posterior mean, or by examining the marginal 95 percent credible intervals.

### Usage

```
nbp.normalmeans(x, a=1/2+1/length(x), b.est=c("fixed", "est.sparsity", "reml"),
                b=1/length(x), sigma2=1, var.select = c("threshold", "intervals"),
                max.steps=10000, burnin=5000)
```

### Arguments

x	an $n \times 1$ multivariate normal vector.
a	The shape parameter $a$ in the $\beta'(a, b)$ prior. Defaults to $1/2 + 1/n$ . User may specify a different value for $a$ ( $a > 0$ ).
b.est	The method for estimating the sparsity parameter $b$ . If "fixed" is chosen, then $b$ is not estimated from the data. If "est.sparsity" is chosen, the empirical Bayes estimate of sparsity level from van der Pas et al. (2014) is used. If "reml" is chosen, $b$ is estimated from restricted marginal maximum likelihood.
b	The shape parameter $b$ in the $\beta'(a, b)$ prior. Controls the sparsity of the model. Defaults to $1/n$ . User may specify a different value ( $b > 0$ ). This is ignored if the method "est.sparsity" or "reml" is used to estimate $b$ .
sigma2	The variance parameter. Defaults to 1. User needs to specify the noise parameter if it is different from 1 ( $\sigma^2 > 0$ ).
var.select	The method of variable selection. "threshold" selects variables by thresholding the shrinkage factor in the posterior mean. "intervals" will classify $\theta_i$ 's as either signals ( $\theta_i \neq 0$ ) or as noise ( $\theta_i = 0$ ) by examining the 95 percent posterior credible intervals.
max.steps	The total number of iterations to run in the Gibbs sampler. Defaults to 10,000.
burnin	The number of burn-in iterations for the Gibbs sampler. Defaults to 5,000.

### Details

The function implements sparse estimation and multiple hypothesis testing on a multivariate normal mean vector,  $\theta = (\theta_1, \dots, \theta_n)$  with the NBP prior of Bai and Ghosh (2018). The full model is:

$$\begin{aligned} X|\theta &\sim N_n(\theta, \sigma^2 I_n), \\ \theta_i|\omega_i^2 &\sim N(0, \sigma^2 \omega_i^2), i = 1, \dots, n, \\ \omega_i^2 &\sim \beta'(a, b), i = 1, \dots, n, \end{aligned}$$

where  $b$  is the main parameter that controls the sparsity of the solution. The sparsity parameter  $b$  can be estimated by the empirical Bayes estimate of the estimated sparsity ("est.sparsity") given in van der Pas et al. (2014), or it can be estimated by restricted marginal maximum likelihood ("rem1") in the interval  $[1/n, 1]$ .

The posterior mean is of the form  $[E(1 - \kappa_i | X_i)] X_i, i = 1, \dots, n$ . The "threshold" method for variable selection is to classify  $\theta_i$  as signal ( $\theta_i \neq 0$ ) if  $E(1 - \kappa_i | X_i) > 1/2$ .

## Value

The function returns a list containing the following components:

theta.hat	The posterior mean estimate of $\theta$ .
theta.med	The posterior median estimate of $\theta$ .
theta.intervals	The 95 percent credible intervals for all $n$ components of $\theta$ .
nbp.classifications	An $n$ -dimensional binary vector with "1" if the covariate is selected and "0" if it is deemed irrelevant.
b.estimate	The estimate of the sparsity level. If user specified "fixed" for b.est, then it simply returns the fixed $b$ .

## Author(s)

Ray Bai and Malay Ghosh

## References

- Bai, R. and Ghosh, M. (2018). "Large-Scale Multiple Hypothesis Testing with the Normal-Beta Prime Prior." Pre-print, arXiv:1807.02421.
- Bai, R. and Ghosh, M. (2018). "On the Beta Prime Prior for Scale Parameters in High-Dimensional Bayesian Regression Models." Pre-print, arXiv: 1807.06539.
- van der Pas, S. L., Kleijn, B. J. K., and van der Vaart, A. W. (2014). "The Horseshoe Estimator: Posterior Concentration Around Nearly Black Vectors." *Electronic Journal of Statistics*, **8**(2):2585-2618.
- van der Pas, S. L., Szabo, B. T., and van der Vaart, A. (2017). "Adaptive posterior contraction rates for the horseshoe." *Electronic Journal of Statistics*, **11**(2):3196-3225.

## Examples

```
#####
# Example on synthetic data. #
# 5 percent of entries in a sparse vector theta #
# are set equal to signal value A = 7. #
#####
n <- 40
sparsity.level <- 5 # 5 percent of entries will be nonzero
A <- 5
```



```

  z.scores[i] <- qnorm(pt(test.stat[i],100))
}

#####
# Apply NBP model on the z-scores. #
# Here sigma2 is known with sigma2= 1 #
#####

nbp.model <- nbp.normalmeans(z.scores, b.est="reml", var.select="threshold")

#####
# How many genes flagged as significant? #
#####
num.sig <- sum(nbp.model$nbpclassifications != 0)
num.sig

#####
# Estimated effect size for 10 most significant genes #
#####
most.sig <- c(610,1720,332,364,914,3940,4546,1068,579,4331)
nbp.model$theta.hat[most.sig]

```

---

singh2002

*Prostate Cancer Study of Singh et al. (2002)*


---

### Description

Gene expression data (6033 genes for 102 samples) from the microarray study of Singh et al. (2002). Also available in the sda package.

### Usage

```
data(singh2002)
```

### Format

A list of two components.

**x:** is a  $102 \times 6033$  matrix containing the expression levels. The rows contain the samples and the columns the genes.

**y:** is a factor containing the diagnosis for each sample ("cancer" or "healthy").

### Details

This data set contains measurements of the gene expression of 6033 genes for 102 observations. The first 52 rows are for the cancer patients and the last 50 rows are for the normal control subjects.

**Source**

The data is described in Singh et al. (2001) and are provided in exactly the form as used by Efron (2010).

**References**

Singh, D., Febbo, P.G., Ross, K., Jackson, D.G., Maonla, J., Ladd, C., Tamayo, P., Renshaw, A.A., D'Amico, A.V., Richie, J.P., Lander, E.S., Loda, M., Kantoff, P.W., Golub, T.R., and Sellers, W.R. (2002). "Gene expression correlates of clinical prostate cancer behavior." *Cancer Cell*, **1**(2):203-209.

Efron, B. (2010). "The Future of Indirect Evidence." *Statistical Science*, **25**(2):145-157.

---

trim32

*Gene expression levels of TRIM32*

---

**Description**

Gene expression levels of TRIM32 from the microarray study by Scheetz et al. (2006). This data is also available in the flare package.

**Usage**

trim32

**Format**

trim32: is a response vector with 120 entries.

**Details**

The data set contains gene expression levels for 120 samples of TRIM32. It comes from the microarray experiments of mammalian eye tissue samples of Scheetz et al. (2006).

**Source**

<https://cran.r-project.org/web/packages/flare/>

**References**

Scheetz, T. E., Kim K.-Y. A., Swiderski R. E., Philp A. R., Braun, T. A., Knudtson, K. L., Dorrance, A. M., DiBona, G. F., Huang, J., Casavant, T. L., Sheffield, V. C., and Stone, E. M. (2006). "Regulation of gene expression in the mammalian eye and its relevance to eye disease." *Proceedings of the National Academy of Sciences*, **103**(39):14429-14434.

# Index

diabetes, [3](#)  
dl.normalmeans, [4](#)  
  
eyedata, [7](#)  
  
genes, [8](#)  
  
hsplus.normalmeans, [9](#)  
  
nbp, [12](#)  
nbp.normalmeans, [16](#)  
NormalBetaPrime  
    ([NormalBetaPrime-package](#)), [2](#)  
[NormalBetaPrime-package](#), [2](#)  
  
singh2002, [20](#)  
  
trim32, [21](#)