Package ‘BayesianLasso’

July 28, 2025

Type Package

Title Bayesian Lasso Regression and Tools for the Lasso Distribution

Version 0.3.5

Date 2025-07-21

Maintainer Mohammad Javad Davoudabadi <mohammad.davoudabadi@sydney.edu.au>

Description Implements Bayesian Lasso regression using efficient Gibbs sampling algorithms, including modified versions of the Hans and Park–Casella (PC) samplers. Includes functions for working with the Lasso distribution, such as its density, cumulative distribution, quantile, and random generation functions, along with moment calculations. Also includes a function to compute the Mills ratio. Designed for sparse linear models and suitable for high-dimensional regression problems.

License GPL-3

Imports Rcpp (>= 1.0.12)

LinkingTo Rcpp, RcppNumerical, RcppArmadillo, RcppEigen, RcppClock

RoxygenNote 7.3.2

Encoding UTF-8

URL https://garthtarr.github.io/BayesianLasso/,
https://github.com/garthtarr/BayesianLasso

VignetteBuilder knitr

Suggests knitr, rmarkdown, monomvn, bayeslm, rstan, bayesreg, lars, Ecdat, testthat (>= 3.0.0), MASS

Config/testthat/edition 3

BugReports https://github.com/garthtarr/BayesianLasso/issues

NeedsCompilation yes

Author John Ormerod [aut, cph] (ORCID: <https://orcid.org/0000-0002-4650-7507>),
Mohammad Javad Davoudabadi [aut, cre, cph] (ORCID: <https://orcid.org/0000-0001-7312-1530>),
Garth Tarr [aut, cph] (ORCID: <https://orcid.org/0000-0002-6605-7478>),
Samuel Mueller [aut, cph] (ORCID:

<https://orcid.org/0000-0002-3087-8127>),
Jonathon Tidswell [aut, cph]

Repository CRAN

Date/Publication 2025-07-28 18:30:07 UTC

Contents

LassoDistribution *The Lasso Distribution*

Description

Provides functions related to the Lasso distribution, including the normalizing constant, probability density function, cumulative distribution function, quantile function, and random number generation for given parameters a, b, and c. Additional utilities include the Mills ratio, expected value, and variance of the distribution. The package also implements modified versions of the Hans and Park–Casella Gibbs sampling algorithms for Bayesian Lasso regression.

Usage

```
zlasso(a, b, c, logarithm)
dlasso(x, a, b, c, logarithm)
plasso(q, a, b, c)
qlasso(p, a, b, c)
rlasso(n, a, b, c)
elasso(a, b, c)
vlasso(a, b, c)
mlasso(a, b, c)
MillsRatio(d)
Modified_Hans_Gibbs(X, y, a1, b1, u1, v1,
              nsamples, beta_init, lambda_init, sigma2_init, verbose)
Modified_PC_Gibbs(X, y, a1, b1, u1, v1,
              nsamples, lambda_init, sigma2_init, verbose)
```

Arguments

x, q	Vector of quantiles (vectorized).
p	Vector of probabilities.
a	Vector of precision parameter which must be non-negative.
b	Vector of off set parameter.
c	Vector of tuning parameter which must be non-negative values.

n	Number of observations.
logarithm	Logical. If TRUE, probabilities are returned on the log scale.
d	A scalar numeric value. Represents the point at which the Mills ratio is evaluated.
X	Design matrix (numeric matrix).
y	Response vector (numeric vector).
a1	Shape parameter of the prior on λ^2.
b1	Rate parameter of the prior on λ^2.
u1	Shape parameter of the prior on σ^2.
v1	Rate parameter of the prior on σ^2.
nsamples	Number of Gibbs samples to draw.
beta_init	Initial value for the model parameter β.
lambda_init	Initial value for the shrinkage parameter λ^2.
sigma2_init	Initial value for the error variance σ^2.
verbose	Integer. If greater than 0, progress is printed every verbose iterations during sampling. Set to 0 to suppress output.

Details

If $X \sim \text{Lasso}(a, b, c)$ then its density function is:

$$p(x; a, b, c) = Z^{-1} \exp\left(-\frac{1}{2}ax^2 + bx - c|x|\right)$$

where $x \in \mathbb{R}$, $a > 0$, $b \in \mathbb{R}$, $c > 0$, and Z is the normalizing constant.

More details are included for the CDF, quantile function, and normalizing constant in the original documentation.

Value

- zlasso, dlasso, plasso, qlasso, rlasso, elasso, vlasso, mlasso, MillsRatio: return the corresponding scalar or vector values related to the Lasso distribution and a numeric value representing the Mills ratio.
- Modified_Hans_Gibbs: returns a list containing:

  mBeta Matrix of MCMC samples for the regression coefficients β, with nsamples rows and p columns.

  vsigma2 Vector of MCMC samples for the error variance σ^2.

  vlambda2 Vector of MCMC samples for the shrinkage parameter λ^2.

  mA Matrix of sampled values for parameter a_j of the Lasso distribution for each β_j.

  mB Matrix of sampled values for parameter b_j of the Lasso distribution for each β_j.

  mC Matrix of sampled values for parameter c_j of the Lasso distribution for each β_j.
- Modified_PC_Gibbs: returns a list containing:

  mBeta Matrix of MCMC samples for the regression coefficients β.

vsigma2 Vector of MCMC samples for the error variance σ^2.

vlambda2 Vector of MCMC samples for the shrinkage parameter λ^2.

mM Matrix of estimated means of the full conditional distributions of each β_j.

mV Matrix of estimated variances of the full conditional distributions of each β_j.

va_til Vector of estimated shape parameters for the full conditional inverse-gamma distribution of σ^2.

vb_til Vector of estimated rate parameters for the full conditional inverse-gamma distribution of σ^2.

vu_til Vector of estimated shape parameters for the full conditional inverse-gamma distribution of λ^2.

vv_til Vector of estimated rate parameters for the full conditional inverse-gamma distribution of λ^2.

See Also

normalize for preprocessing input data before applying the samplers.

Examples

```
a <- 2; b <- 1; c <- 3
x <- seq(-3, 3, length.out = 1000)
plot(x, dlasso(x, a, b, c, logarithm = FALSE), type = 'l')

r <- rlasso(1000, a, b, c)
hist(r, breaks = 50, probability = TRUE, col = "grey", border = "white")
lines(x, dlasso(x, a, b, c, logarithm = FALSE), col = "blue")

plasso(0, a, b, c)
qlasso(0.25, a, b, c)
elasso(a, b, c)
vlasso(a, b, c)
mlasso(a, b, c)
MillsRatio(2)



# The Modified_Hans_Gibbs() function uses the Lasso distribution to draw
# samples from the full conditional distribution of the regression coefficients.

y <- 1:20
X <- matrix(c(1:20,12:31,7:26),20,3,byrow = TRUE)

a1 <- b1 <- u1 <- v1 <- 0.01
sigma2_init <- 1
lambda_init <- 0.1
beta_init <- rep(1, ncol(X))
nsamples <- 1000
verbose <- 100

Output_Hans <- Modified_Hans_Gibbs(
```

```
                X, y, a1, b1, u1, v1,
                nsamples, beta_init, lambda_init, sigma2_init, verbose
)

colMeans(Output_Hans$mBeta)
mean(Output_Hans$vlambda2)


Output_PC <- Modified_PC_Gibbs(
               X, y, a1, b1, u1, v1,
               nsamples, lambda_init, sigma2_init, verbose)

colMeans(Output_PC$mBeta)
mean(Output_PC$vlambda2)
```

`normalize` *Normalize Response and Covariates*

Description

This function centers and (optionally) scales the response vector and each column of the design matrix using the population variance. It is used to prepare data for Bayesian Lasso regression.

Usage

```
normalize(y, X, scale = TRUE)
```

Arguments

`y`	A numeric response vector.
`X`	A numeric matrix or data frame of covariates (design matrix).
`scale`	Logical; if `TRUE`, variables are scaled to have unit population variance (default is `TRUE`).

Value

A list with the following elements:

- `vy`: Normalized response vector.
- `mX`: Normalized design matrix.
- `mu.y`: Mean of the response vector.
- `sigma2.y`: Population variance of the response vector.
- `mu.x`: Vector of column means of `X`.
- `sigma2.x`: Vector of population variances for columns of `X`.

Examples

```
set.seed(1)
X <- matrix(rnorm(100 * 10), 100, 10)
beta <- c(2, -3, rep(0, 8))
y <- as.vector(X %*% beta + rnorm(100))
norm_result <- normalize(y, X)
```

Index