Package: Mhorseshoe (via r-universe)

August 27, 2024

Title Approximate Algorithm for Horseshoe Prior

Version 0.1.3

Description Provides exact and approximate algorithms for the horseshoe prior in linear regression models, which were proposed by Johndrow et al. (2020) <https://www.jmlr.org/papers/v21/19-536.html>.

Encoding UTF-8

Imports stats, Rcpp (>= 1.0.11)

LinkingTo Rcpp

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

License MIT + file LICENSE

Suggests knitr, rmarkdown, ggplot2, horseshoe, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Repository https://rkdalsrl1511.r-universe.dev

RemoteUrl https://github.com/rkdalsrl1511/mhorseshoe

RemoteRef HEAD

RemoteSha fd06a0dba391ce2dbce21cca35adcaf2f9c072d5

Contents

`approx_horseshoe` *Run approximate MCMC algorithm for horseshoe prior*

Description

The approximate MCMC algorithm for the horseshoe prior

Usage

```
approx_horseshoe(
  y,
  X,
  burn = 1000,
  iter = 5000,
  auto.threshold = TRUE,
  threshold = 0,
  tau = 1,
  s = 0.8,
  sigma2 = 1,
  w = 1,
  alpha = 0.05,
  a = 0.2,
  b = 10,
  t = 10,
  adapt_p0 = 0,
  adapt_p1 = -4.6 * 10^(-4)
)
```

Arguments

`y`	Response vector, $y \in \mathbb{R}^N$.
`X`	Design matrix, $X \in \mathbb{R}^{N \times p}$.
`burn`	Number of burn-in samples. The default is 1000.
`iter`	Number of samples to be drawn from the posterior. The default is 5000.
`auto.threshold`	Argument for setting whether to use an algorithm that automatically updates the threshold using adaptive probability.
`threshold`	Threshold to be used in the approximate MCMC algorithm. This argument is ignored when auto.threshold=TRUE. If you select auto.threshold = FALSE and threshold = 0 (This is the default value for the threshold argument), the threshold is set to $\sqrt{p \times min(N, p)}$ as suggested in Johndrow et al. (2020). Or, you can set your custom value directly through this argument. For more information about δ, browseVignettes("Mhorseshoe") and 4.1 of Johndrow et al. (2020).
`tau`	Initial value of the global shrinkage parameter τ when starting the algorithm. The default is 1.

s	s^2 is the variance of tau's MH proposal distribution. 0.8 is a good default. If set to 0, the algorithm proceeds by fixing the global shrinkage parameter τ to the initial setting value.
sigma2	Initial value of error variance σ^2. The default is 1.
w	A hyperparameter of gamma prior for σ^2. The default is 1.
alpha	$100(1-\alpha)\%$ credible interval setting argument.
a	A tuning parameter of the rejection sampler, where the default value is $a = 1/5$.
b	A tuning parameter of the rejection sampler, where the default value is $b = 10$.
t	Threshold update cycle for adaptive probability algorithm when auto.threshold is set to TRUE. The default is 10.
adapt_p0	A tuning parameter p_0 of the adaptive probability, $p(t) = exp[p_0 + p_1 t]$. The default is 0.
adapt_p1	A tuning parameter a_1 of the adaptive probability, $p(t) = exp[p_0 + p_1 t]$. The default is -4.6×10^{-4}.

Details

This function implements the approximate algorithm introduced in Section 2.2 of Johndrow et al. (2020) and the method proposed in this package, which improves computation speed when p » N. The approximate algorithm introduces a threshold and uses only a portion of the total p columns for matrix multiplication, reducing the computational cost compared to the existing MCMC algorithms for the horseshoe prior. The "auto.threshold" argument determines whether the threshold used in the algorithm will be updated by the adaptive method proposed in this package.

Value

BetaHat	Posterior mean of β.
LeftCI	Lower bound of $100(1-\alpha)\%$ credible interval for β.
RightCI	Upper bound of $100(1-\alpha)\%$ credible interval for β.
Sigma2Hat	Posterior mean of σ^2.
TauHat	Posterior mean of τ.
LambdaHat	Posterior mean of $\lambda_j,\ j = 1, 2, ...p..$
ActiveMean	Average number of elements in the active set per iteration in this algorithm.
BetaSamples	Posterior samples of β.
LambdaSamples	Posterior samples of local shrinkage parameters.
TauSamples	Posterior samples of global shrinkage parameter.
Sigma2Samples	Posterior samples of $sigma^2$.
ActiveSet	$\mathbb{R}^{iter \times p}$ Matrix indicating active elements as 1 and non-active elements as 0 per iteration of the MCMC algorithm.

References

Johndrow, J., Orenstein, P., & Bhattacharya, A. (2020). Scalable Approximate MCMC Algorithms for the Horseshoe Prior. In Journal of Machine Learning Research, 21, 1-61.

Examples

```
# Making simulation data.
set.seed(123)
N <- 200
p <- 100
true_beta <- c(rep(1, 10), rep(0, 90))

X <- matrix(1, nrow = N, ncol = p) # Design matrix X.
for (i in 1:p) {
  X[, i] <- stats::rnorm(N, mean = 0, sd = 1)
}

y <- vector(mode = "numeric", length = N) # Response variable y.
e <- rnorm(N, mean = 0, sd = 2) # error term e.
for (i in 1:10) {
  y <- y + true_beta[i] * X[, i]
}
y <- y + e

# Run with auto.threshold set to TRUE
result1 <- approx_horseshoe(y, X, burn = 0, iter = 100,
                            auto.threshold = TRUE)

# Run with fixed custom threshold
result2 <- approx_horseshoe(y, X, burn = 0, iter = 100,
                            auto.threshold = FALSE, threshold = 1/(5 * p))

# posterior mean
betahat <- result1$BetaHat

# Lower bound of the 95% credible interval
leftCI <- result1$LeftCI

# Upper bound of the 95% credible interval
RightCI <- result1$RightCI
```

`exact_horseshoe` *Run exact MCMC algorithm for horseshoe prior*

Description

The exact MCMC algorithm for the horseshoe prior introduced in section 2.1 of Johndrow et al. (2020).

Usage

```
exact_horseshoe(
  y,
```

```
  X,
  burn = 1000,
  iter = 5000,
  a = 1/5,
  b = 10,
  s = 0.8,
  tau = 1,
  sigma2 = 1,
  w = 1,
  alpha = 0.05
)
```

Arguments

y	Response vector, $y \in \mathbb{R}^N$.
X	Design matrix, $X \in \mathbb{R}^{N \times p}$.
burn	Number of burn-in samples. The default is 1000.
iter	Number of samples to be drawn from the posterior. The default is 5000.
a	A tuning parameter of the rejection sampler, where the default value is $a = 1/5$.
b	A tuning parameter of the rejection sampler, where the default value is $b = 10$.
s	s^2 is the variance of tau's MH proposal distribution. 0.8 is a good default. If set to 0, the algorithm proceeds by fixing the global shrinkage parameter τ to the initial setting value.
tau	Initial value of the global shrinkage parameter τ when starting the algorithm. The default is 1.
sigma2	Initial value of error variance σ^2. The default is 1.
w	A hyperparameter of gamma prior for σ^2. The default is 1.
alpha	$100(1-\alpha)\%$ credible interval setting argument.

Details

The exact MCMC algorithm introduced in Section 2.1 of Johndrow et al. (2020) is implemented in this function. This algorithm uses a blocked-Gibbs sampler for (τ, β, σ^2), where the global shrinkage parameter τ is updated by an Metropolis-Hastings algorithm. The local shrinkage parameter $\lambda_j,\ j = 1, 2, ..., p$ is updated by the rejection sampler.

Value

BetaHat	Posterior mean of β.
LeftCI	Lower bound of $100(1-\alpha)\%$ credible interval for β.
RightCI	Upper bound of $100(1-\alpha)\%$ credible interval for β.
Sigma2Hat	Posterior mean of σ^2.
TauHat	Posterior mean of τ.
LambdaHat	Posterior mean of $\lambda_j,\ j = 1, 2, ...p$..

BetaSamples	Samples from the posterior of β.
LambdaSamples	Lambda samples through rejection sampling.
TauSamples	Tau samples through MH algorithm.
Sigma2Samples	Samples from the posterior of the parameter $sigma^2$.

References

Johndrow, J., Orenstein, P., & Bhattacharya, A. (2020). Scalable Approximate MCMC Algorithms for the Horseshoe Prior. In Journal of Machine Learning Research, 21, 1-61.

Examples

```
# Making simulation data.
set.seed(123)
N <- 50
p <- 100
true_beta <- c(rep(1, 10), rep(0, 90))

X <- matrix(1, nrow = N, ncol = p) # Design matrix X.
for (i in 1:p) {
  X[, i] <- stats::rnorm(N, mean = 0, sd = 1)
}

y <- vector(mode = "numeric", length = N) # Response variable y.
e <- rnorm(N, mean = 0, sd = 2) # error term e.
for (i in 1:10) {
  y <- y + true_beta[i] * X[, i]
}
y <- y + e

# Run exact_horseshoe
result <- exact_horseshoe(y, X, burn = 0, iter = 100)

# posterior mean
betahat <- result$BetaHat

# Lower bound of the 95% credible interval
leftCI <- result$LeftCI

# Upper bound of the 95% credible interval
RightCI <- result$RightCI
```

Index