Package ‘horseshoe’

July 18, 2019

Title Implementation of the Horseshoe Prior

Version 0.2.0

Description Contains functions for applying the horseshoe prior to high-dimensional linear regression, yielding the posterior mean and credible intervals, amongst other things. The key parameter tau can be equipped with a prior or estimated via maximum marginal likelihood estimation (MMLE). The main function, horseshoe, is for linear regression. In addition, there are functions specifically for the sparse normal means problem, allowing for faster computation of for example the posterior mean and posterior variance. Finally, there is a function available to perform variable selection, using either a form of thresholding, or credible intervals.

Depends R (>= 3.1.0)

Imports stats

Suggests Hmisc, ggplot2, knitr, rmarkdown

Encoding UTF-8

License GPL-3

LazyData false

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Stephanie van der Pas [cre, aut],
James Scott [aut],
Antik Chakraborty [aut],
Anirban Bhattacharya [aut]

Maintainer Stephanie van der Pas <svdpas@math.leidenuniv.nl>

Repository CRAN

Date/Publication 2019-07-18 10:14:05 UTC

R topics documented:

`horseshoe` *Function to implement the horseshoe shrinkage prior in Bayesian linear regression*

Description

This function employs the algorithm proposed in Bhattacharya et al. (2016). The global-local scale parameters are updated via a slice sampling scheme given in the online supplement of Polson et al. (2014). Two different algorithms are used to compute posterior samples of the $p * 1$ vector of regression coefficients β. The method proposed in Bhattacharya et al. (2016) is used when $p > n$, and the algorithm provided in Rue (2001) is used for the case $p <= n$. The function includes options for full hierarchical Bayes versions with hyperpriors on all parameters, or empirical Bayes versions where some parameters are taken equal to a user-selected value.

Usage

```
horseshoe(y, X, method.tau = c("fixed", "truncatedCauchy", "halfCauchy"),
  tau = 1, method.sigma = c("fixed", "Jeffreys"), Sigma2 = 1,
  burn = 1000, nmc = 5000, thin = 1, alpha = 0.05)
```

Arguments

`y`	Response, a $n * 1$ vector.
`X`	Matrix of covariates, dimension $n * p$.
`method.tau`	Method for handling τ. Select "truncatedCauchy" for full Bayes with the Cauchy prior truncated to [1/p, 1], "halfCauchy" for full Bayes with the half-Cauchy prior, or "fixed" to use a fixed value (an empirical Bayes estimate, for example).
`tau`	Use this argument to pass the (estimated) value of τ in case "fixed" is selected for method.tau. Not necessary when method.tau is equal to"halfCauchy" or "truncatedCauchy". The default (tau = 1) is not suitable for most purposes and should be replaced.
`method.sigma`	Select "Jeffreys" for full Bayes with Jeffrey's prior on the error variance σ^2, or "fixed" to use a fixed value (an empirical Bayes estimate, for example).
`Sigma2`	A fixed value for the error variance σ^2. Not necessary when method.sigma is equal to "Jeffreys". Use this argument to pass the (estimated) value of Sigma2 in case "fixed" is selected for method.sigma. The default (Sigma2 = 1) is not suitable for most purposes and should be replaced.
`burn`	Number of burn-in MCMC samples. Default is 1000.
`nmc`	Number of posterior draws to be saved. Default is 5000.

`thin` Thinning parameter of the chain. Default is 1 (no thinning).

`alpha` Level for the credible intervals. For example, alpha = 0.05 results in 95% credible intervals.

Details

The model is:

$$y = X\beta + \epsilon, \epsilon \sim N(0, \sigma^2)$$

The full Bayes version of the horseshoe, with hyperpriors on both τ and σ^2 is:

$$\beta_j \sim N(0, \sigma^2 \lambda_j^2 \tau^2)$$

$$\lambda_j \sim Half - Cauchy(0, 1), \tau \sim Half - Cauchy(0, 1)$$

$$\sigma^2 \sim 1/\sigma^2$$

There is an option for a truncated Half-Cauchy prior (truncated to [1/p, 1]) on τ. Empirical Bayes versions are available as well, where τ and/or σ^2 are taken equal to fixed values, possibly estimated using the data.

Value

`BetaHat` Posterior mean of Beta, a p by 1 vector.

`LeftCI` The left bounds of the credible intervals.

`RightCI` The right bounds of the credible intervals.

`BetaMedian` Posterior median of Beta, a p by 1 vector.

`Sigma2Hat` Posterior mean of error variance σ^2. If method.sigma = "fixed" is used, this value will be equal to the user-selected value of Sigma2 passed to the function.

`TauHat` Posterior mean of global scale parameter tau, a positive scalar. If method.tau = "fixed" is used, this value will be equal to the user-selected value of tau passed to the function.

`BetaSamples` Posterior samples of Beta.

`TauSamples` Posterior samples of tau.

`Sigma2Samples` Posterior samples of Sigma2.

References

Bhattacharya A., Chakraborty A., and Mallick B.K (2016), Fast sampling with Gaussian scale-mixture priors in high-dimensional regression. Biometrika 103(4), 985–991.

Polson, N.G., Scott, J.G. and Windle, J. (2014) The Bayesian Bridge. Journal of Royal Statistical Society, B, 76(4), 713-733.

Rue, H. (2001). Fast sampling of Gaussian Markov random fields. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63, 325–338.

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

See Also

`HS.normal.means` for a faster version specifically for the sparse normal means problem (design matrix X equal to identity matrix) and `HS.post.mean` for a fast way to estimate the posterior mean in the sparse normal means problem when a value for tau is available.

Examples

```
## Not run: #In this example, there are no relevant predictors
#20 observations, 30 predictors (betas)
y <- rnorm(20)
X <- matrix(rnorm(20*30) , 20)
res <- horseshoe(y, X, method.tau = "truncatedCauchy", method.sigma = "Jeffreys")

plot(y, X%*%res$BetaHat) #plot predicted values against the observed data
res$TauHat #posterior mean of tau
HS.var.select(res, y, method = "intervals") #selected betas
#Ideally, none of the betas is selected (all zeros)
#Plot the credible intervals
library(Hmisc)
xYplot(Cbind(res$BetaHat, res$LeftCI, res$RightCI) ~ 1:30)

## End(Not run)

## Not run:  #The horseshoe applied to the sparse normal means problem
# (note that HS.normal.means is much faster in this case)
X <- diag(100)
beta <- c(rep(0, 80), rep(8, 20))
y <- beta + rnorm(100)
res2 <- horseshoe(y, X, method.tau = "truncatedCauchy", method.sigma = "Jeffreys")
#Plot predicted values against the observed data (signals in blue)
plot(y, X%*%res2$BetaHat, col = c(rep("black", 80), rep("blue", 20)))
res2$TauHat #posterior mean of tau
HS.var.select(res2, y, method = "intervals") #selected betas
#Ideally, the final 20 predictors are selected
#Plot the credible intervals
library(Hmisc)
xYplot(Cbind(res2$BetaHat, res2$LeftCI, res2$RightCI) ~ 1:100)

## End(Not run)
```

`HS.MMLE` *MMLE for the horseshoe prior for the sparse normal means problem.*

Description

Compute the marginal maximum likelihood estimator (MMLE) of tau for the horseshoe for the normal means problem (i.e. linear regression with the design matrix equal to the identity matrix). The MMLE is explained and studied in Van der Pas et al. (2016).

Usage

```
HS.MMLE(y, Sigma2)
```

Arguments

y	The data, a $n * 1$ vector.
Sigma2	The variance of the data.

Details

The normal means model is:

$$y_i = \beta_i + \epsilon_i, \epsilon_i \sim N(0, \sigma^2)$$

And the horseshoe prior:

$$\beta_j \sim N(0, \sigma^2 \lambda_j^2 \tau^2)$$

$$\lambda_j \sim Half - Cauchy(0, 1).$$

This function estimates τ. A plug-in value of σ^2 is used.

Value

The MMLE for the parameter tau of the horseshoe.

Note

Requires a minimum of 2 observations. May return an error for vectors of length larger than 400 if the truth is very sparse. In that case, try `HS.normal.means`.

References

van der Pas, S.L., Szabo, B., and van der Vaart, A. (2017), Uncertainty quantification for the horseshoe (with discussion). Bayesian Analysis 12(4), 1221-1274.

van der Pas, S.L., Szabo, B., and van der Vaart A. (2017), Adaptive posterior contraction rates for the horseshoe. Electronic Journal of Statistics 10(1), 3196-3225.

See Also

The estimated value of τ can be plugged into `HS.post.mean` to obtain the posterior mean, and into `HS.post.var` to obtain the posterior variance. These functions are all for empirical Bayes; if a full Bayes version with a hyperprior on τ is preferred, see `HS.normal.means` for the normal means problem, or `horseshoe` for linear regression.

Examples

```
## Not run: #Example with 5 signals, rest is noise
truth <- c(rep(0, 95), rep(8, 5))
y <-  truth + rnorm(100)
(tau.hat <- HS.MMLE(y, 1)) #returns estimate of tau
plot(y, HS.post.mean(y, tau.hat, 1)) #plot estimates against the data

## End(Not run)
## Not run: #Example where the data variance is estimated first
truth <- c(rep(0, 950), rep(8, 50))
y <-  truth + rnorm(100, mean = 0, sd = sqrt(2))
sigma2.hat <- var(y)
(tau.hat <- HS.MMLE(y, sigma2.hat)) #returns estimate of tau
plot(y, HS.post.mean(y, tau.hat, sigma2.hat)) #plot estimates against the data

## End(Not run)
```

`HS.normal.means` *The horseshoe prior for the sparse normal means problem*

Description

Apply the horseshoe prior to the normal means problem (i.e. linear regression with the design matrix equal to the identity matrix). Computes the posterior mean, median and credible intervals. There are options for empirical Bayes (estimate of tau and or Sigma2 plugged in) and full Bayes (truncated or non-truncated half-Cauchy on tau, Jeffrey's prior on Sigma2). For the full Bayes version, the truncated half-Cauchy prior is recommended by Van der Pas et al. (2016).

Usage

```
HS.normal.means(y, method.tau = c("fixed", "truncatedCauchy",
  "halfCauchy"), tau = 1, method.sigma = c("fixed", "Jeffreys"),
  Sigma2 = 1, burn = 1000, nmc = 5000, alpha = 0.05)
```

Arguments

`y`	The data. A $n * 1$ vector.
`method.tau`	Method for handling τ. Select "fixed" to plug in an estimate of tau (empirical Bayes), "truncatedCauchy" for the half- Cauchy prior truncated to [1/n, 1], or "halfCauchy" for a non-truncated half-Cauchy prior. The truncated Cauchy prior is recommended over the non-truncated version.
`tau`	Use this argument to pass the (estimated) value of τ in case "fixed" is selected for method.tau. Not necessary when method.tau is equal to"halfCauchy" or "truncatedCauchy". The function `HS.MMLE` can be used to compute an estimate of tau. The default (tau = 1) is not suitable for most purposes and should be replaced.

`method.sigma`	Select "fixed" for a fixed error variance, or "Jeffreys" to use Jeffrey's prior.
`Sigma2`	The variance of the data - only necessary when "fixed" is selected for method.sigma. The default (Sigma2 = 1) is not suitable for most purposes and should be replaced.
`burn`	Number of samples used for burn-in. Default is 1000.
`nmc`	Number of MCMC samples taken after burn-in. Default is 5000.
`alpha`	The level for the credible intervals. E.g. alpha = 0.05 yields 95% credible intervals

Details

The normal means model is:

$$y_i = \beta_i + \epsilon_i, \epsilon_i \sim N(0, \sigma^2)$$

And the horseshoe prior:

$$\beta_j \sim N(0, \sigma^2 \lambda_j^2 \tau^2)$$

$$\lambda_j \sim Half - Cauchy(0, 1).$$

Estimates of τ and σ^2 may be plugged in (empirical Bayes), or those parameters are equipped with hyperpriors (full Bayes).

Value

`BetaHat`	The posterior mean (horseshoe estimator) for each of the datapoints.
`LeftCI`	The left bounds of the credible intervals.
`RightCI`	The right bounds of the credible intervals.
`BetaMedian`	Posterior median of Beta, a n by 1 vector.
`Sigma2Hat`	Posterior mean of error variance σ^2. If method.sigma = "fixed" is used, this value will be equal to the user-selected value of Sigma2 passed to the function.
`TauHat`	Posterior mean of global scale parameter tau, a positive scalar. If method.tau = "fixed" is used, this value will be equal to the user-selected value of tau passed to the function.
`BetaSamples`	Posterior samples of Beta.
`TauSamples`	Posterior samples of tau.
`Sigma2Samples`	Posterior samples of Sigma2.

References

van der Pas, S.L., Szabo, B., and van der Vaart, A. (2017), Uncertainty quantification for the horseshoe (with discussion). Bayesian Analysis 12(4), 1221-1274.

van der Pas, S.L., Szabo, B., and van der Vaart A. (2017), Adaptive posterior contraction rates for the horseshoe. Electronic Journal of Statistics 10(1), 3196-3225.

See Also

`HS.post.mean` for a fast way to compute the posterior mean if an estimate of tau is available. `horseshoe` for linear regression. `HS.var.select` to perform variable selection.

Examples

```
#Empirical Bayes example with 20 signals, rest is noise
#Posterior mean for the signals is plotted
#And variable selection is performed using the credible intervals
#And the credible intervals are plotted
truth <- c(rep(0, 80), rep(8, 20))
data <-  truth + rnorm(100, 1)
tau.hat <- HS.MMLE(data, Sigma2 = 1)
res.HS1 <- HS.normal.means(data, method.tau = "fixed", tau = tau.hat,
method.sigma = "fixed", Sigma2 = 1)
#Plot the posterior mean against the data (signals in blue)
plot(data, res.HS1$BetaHat, col = c(rep("black", 80), rep("blue", 20)))
#Find the selected betas (ideally, the last 20 are equal to 1)
HS.var.select(res.HS1, data, method = "intervals")
#Plot the credible intervals
library(Hmisc)
xYplot(Cbind(res.HS1$BetaHat, res.HS1$LeftCI, res.HS1$RightCI) ~ 1:100)


#Full Bayes example with 20 signals, rest is noise
#Posterior mean for the signals is plotted
#And variable selection is performed using the credible intervals
#And the credible intervals are plotted
truth <- c(rep(0, 80), rep(8, 20))
data <-  truth + rnorm(100, 3)
res.HS2 <- HS.normal.means(data, method.tau = "truncatedCauchy", method.sigma = "Jeffreys")
#Plot the posterior mean against the data (signals in blue)
plot(data, res.HS2$BetaHat, col = c(rep("black", 80), rep("blue", 20)))
#Find the selected betas (ideally, the last 20 are equal to 1)
HS.var.select(res.HS2, data, method = "intervals")
#Plot the credible intervals
library(Hmisc)
xYplot(Cbind(res.HS2$BetaHat, res.HS2$LeftCI, res.HS2$RightCI) ~ 1:100)
```

`HS.post.mean` *Posterior mean for the horseshoe for the normal means problem.*

Description

Compute the posterior mean for the horseshoe for the normal means problem (i.e. linear regression with the design matrix equal to the identity matrix), for a fixed value of tau, without using MCMC, leading to a quick estimate of the underlying parameters (betas). Details on computation are given in Carvalho et al. (2010) and Van der Pas et al. (2014).

Usage

```
HS.post.mean(y, tau, Sigma2 = 1)
```

Arguments

`y`	The data. An $n * 1$ vector.
`tau`	Value for tau. Warning: tau should be greater than 1/450.
`Sigma2`	The variance of the data.

Details

The normal means model is:

$$y_i = \beta_i + \epsilon_i, \epsilon_i \sim N(0, \sigma^2)$$

And the horseshoe prior:

$$\beta_j \sim N(0, \sigma^2 \lambda_j^2 \tau^2)$$

$$\lambda_j \sim Half - Cauchy(0, 1).$$

If τ and σ^2 are known, the posterior mean can be computed without using MCMC.

Value

The posterior mean (horseshoe estimator) for each of the datapoints.

References

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010), The horseshoe estimator for sparse signals. Biometrika 97(2), 465–480.

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.

See Also

`HS.post.var` to compute the posterior variance. See `HS.normal.means` for an implementation that does use MCMC, and returns credible intervals as well as the posterior mean (and other quantities). See `horseshoe` for linear regression.

Examples

```
#Plot the posterior mean for a range of deterministic values
y <- seq(-5, 5, 0.05)
plot(y, HS.post.mean(y, tau = 0.5, Sigma2 = 1))

#Example with 20 signals, rest is noise
#Posterior mean for the signals is plotted in blue
truth <- c(rep(0, 80), rep(8, 20))
data <-  truth + rnorm(100)
tau.example <- HS.MMLE(data, 1)
```

```
plot(data, HS.post.mean(data, tau.example, 1),
 col = c(rep("black", 80), rep("blue", 20)))
```

`HS.post.var` *Posterior variance for the horseshoe for the normal means problem.*

Description

Compute the posterior variance for the horseshoe for the normal means problem (i.e. linear regression with the design matrix equal to the identity matrix), for a fixed value of tau, without using MCMC. Details on computation are given in Carvalho et al. (2010) and Van der Pas et al. (2014).

Usage

```
HS.post.var(y, tau, Sigma2)
```

Arguments

`y`	The data. An $n * 1$ vector.
`tau`	Value for tau. Tau should be greater than 1/450.
`Sigma2`	The variance of the data.

Details

The normal means model is:

$$y_i = \beta_i + \epsilon_i, \epsilon_i \sim N(0, \sigma^2)$$

And the horseshoe prior:

$$\beta_j \sim N(0, \sigma^2 \lambda_j^2 \tau^2)$$

$$\lambda_j \sim Half - Cauchy(0, 1).$$

If τ and σ^2 are known, the posterior variance can be computed without using MCMC.

Value

The posterior variance for each of the datapoints.

References

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010), The horseshoe estimator for sparse signals. Biometrika 97(2), 465–480.

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.

See Also

`HS.post.mean` to compute the posterior mean. See `HS.normal.means` for an implementation that does use MCMC, and returns credible intervals as well as the posterior mean (and other quantities). See `horseshoe` for linear regression.

Examples

```
#Plot the posterior variance for a range of deterministic values
y <- seq(-8, 8, 0.05)
plot(y, HS.post.var(y, tau = 0.05, Sigma2 = 1))

#Example with 20 signals, rest is noise
#Posterior variance for the signals is plotted in blue
#Posterior variance for the noise is plotted in black
truth <- c(rep(0, 80), rep(8, 20))
data <-  truth + rnorm(100)
tau.example <- HS.MMLE(data, 1)
plot(data, HS.post.var(data, tau.example, 1),
 col = c(rep("black", 80), rep("blue", 20)) )
```

`HS.var.select` *Variable selection using the horseshoe prior*

Description

The function implements two methods to perform variable selection. The first checks whether 0 is contained in the credible set (see Van der Pas et al. (2016)). The second is only intended for the sparse normal means problem (regression with identity matrix). It is described in Carvalho et al. (2010). The horseshoe posterior mean can be written as $c_i y_i$, with y_i the observation. A variable is selected if $c_i \geq c$, where c is a user-specified threshold.

Usage

```
HS.var.select(hsobject, y, method = c("intervals", "threshold"),
  threshold = 0.5)
```

Arguments

`hsobject`	The outcome from one of the horseshoe functions `horseshoe` or `HS.normal.means`.
`y`	The data.
`method`	Use "intervals" to perform variable selection using the credible sets (at the level specified when creating the hsobject), "threshold" to perform variable selection using the thresholding procedure (only for the sparse normal means problem).
`threshold`	Threshold for the thresholding procedure. Default is 0.5.

Value

A vector of zeroes and ones. The ones correspond to the selected variables.

References

van der Pas, S.L., Szabo, B., and van der Vaart, A. (2017), Uncertainty quantification for the horseshoe (with discussion). Bayesian Analysis 12(4), 1221-1274.

van der Pas, S.L., Szabo, B., and van der Vaart A. (2017), Adaptive posterior contraction rates for the horseshoe. Electronic Journal of Statistics 10(1), 3196-3225.

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

See Also

`horseshoe` and `HS.normal.means` to obtain the required hsobject.

Examples

```
#Example with 20 signals (last 20 entries), rest is noise
truth <- c(rep(0, 80), rep(8, 20))
data <-  truth + rnorm(100)
horseshoe.results <- HS.normal.means(data, method.tau = "truncatedCauchy",
 method.sigma = "fixed")
#Using credible sets. Ideally, the first 80 entries are equal to 0,
#and the last 20 entries equal to 1.
HS.var.select(horseshoe.results, data, method = "intervals")
#Using thresholding. Ideally, the first 80 entries are equal to 0,
#and the last 20 entries equal to 1.
HS.var.select(horseshoe.results, data, method = "threshold")
```