Package ‘localrhat’

December 9, 2021

Type Package

Title A Local Version of R-hat For MCMC Convergence Diagnostic

Version 0.1.0

Maintainer Théo Moins <theo.moins@inria.fr>

Description Implementation of a modified version of the Gelman-Rubin diagnostic for
MCMC convergence diagnostic (also known as R-hat), where it is
computed on local indicator variables and the supremum over all
quantile is consider.

Depends R (>= 4.1), MASS

License What license is it under?

Encoding UTF-8

RoxygenNote 7.1.2

R topics documented:

all_local_rhat *Computation of $\hat{R}(x)$ on a given grid*

Description

Return the vector of $\hat{R}(x)$ values that will be used to compute the supremum $\hat{R}_\infty$. The function use grid_for_R for the grid of values of x.

Usage

```
all_local_rhat(chains, max_nb_points = 500)
```

Arguments

chains	an array of size $n \times m$ where n is the length of the chains and $m \geq 2$ is the number of chains.
max_nb_points	the maximal length of the grid in the case where the total number of samples is larger. By default, max_nb_points = 500.

Details

The computation of $\hat{R}_\infty$ require to compute a supremum of $\hat{R}(x)$ values over the quantiles x. See the documentation of grid_for_R to see how the choice of the different x is done. Thus, this function combine grid_for_R for the computation of the grid and local_rhat for the computation of $\hat{R}(x)$ on a given x to obtain all the values of $\hat{R}(x)$ on the generated grid.

Value

A list that contains the different $\hat{R}(x)$ values used for the computation of $\hat{R}_\infty$.

References

TO INCLUDE

Examples

```
library(localrhat)

N <- 500 # length of chains
M <- 4 # number of chains

# Toy example with 3 i.i.d chains uniform in [-0.5, 0.5] and 1 in [-1, 1]:
chains <- array(c(runif((M-1)*N, -1/2, 1/2), runif(N, -1,1)), c(N,M))

# plot_local_r(grid_for_R(chains), all_local_rhat(chains),
#              xlim = c(-1, 1), ylim=c(0.999,1.1), title ="Gaussian distributions")
```

grid_for_R *Quantile values used for the computation of $\hat{R}$_∞*

Description

Return the set of points that will be used to estimate the supremum of $\hat{R}(x)$ over x. The number of different values taken by $\hat{R}(x)$ can not exceed the number of samples, so the number of points used for the computation is the minimum between $n \times m$ and a threshold value that can be specified in the argument `max_nb_points`

Usage

```
grid_for_R(chains, max_nb_points = 500)
```

Arguments

`chains`	an array of size $n \times m$ where n is the length of the chains and $m \geq 2$ is the number of chains.
`max_nb_points`	the maximal length of the grid in the case where the total number of samples is larger. By default, `max_nb_points = 500`.

Details

As the computation of local-$\hat{R}$ are based on indicator values $I(\theta^{(i,j)} \leq x)$ and because the chains are finite, it is enough to evaluate these indicators in $x = \theta^{(i',j')}$ to obtain all the possible values of $\hat{R}(x)$. However, to keep the computation time reasonable, if $n \times m$ is greater than a given `max_nb_points`, the values are sorted and uniform thinning is apply (which means picking off every k iterations) so that only `max_nb_points` is kept at the end.

Value

A list that contains the different x used for the computation of $\hat{R}(x)$.

References

TO INCLUDE

Examples

```
library(localrhat)

N <- 500 # length of chains
M <- 4 # number of chains

# Toy example with 3 i.i.d chains uniform in [-0.5, 0.5] and 1 in [-1, 1]:
chains <- array(c(runif((M-1)*N, -1/2, 1/2), runif(N, -1,1)), c(N,M))

# plot_local_r(grid_for_R(chains), all_local_rhat(chains),
#             xlim = c(-1, 1), ylim=c(0.999,1.1), title ="Gaussian distributions")
```

`local_rhat` *Local version of the Gelman Rubin diagnostic $\hat{R}$*

Description

Compute $\hat{R}(x)$, a version of univariate $\hat{R}$ computed on indicator variables for a given quantile x.

Usage

```
local_rhat(x, chains)
```

Arguments

`x` a float number corresponding to the quantile used for the computation of $\hat{R}(x)$.

`chains` an array of size $n \times m$ where n is the length of the chains and $m \geq 2$ is the number of chains.

Details

The function return $\hat{R}$ computed on $I(\theta^{(i,j)} \leq x)$ for a given value of x:

$$\hat{R}(x) = \sqrt{\frac{\frac{n-1}{n}\hat{W}(x) + \hat{B}(x)}{\hat{W}(x)}},$$

with $\hat{W}(x)$ and $\hat{B}(x)$ the estimated local within-chain variance and between-chain variance:

$$\hat{W}(x) = \frac{1}{m}\sum_{j=1}^{m}(F_j(x) - F_j^2(x))$$

And

$$\hat{B}(x) = \frac{1}{m^2}\sum_{j<k}\left(F_j(x) - F_k(x)\right)^2.$$

Value

The local-$\hat{R}(x)$ computed on the m chains.

References

TO INCLUDE

Examples

```
library(localrhat)

N <- 500 # length of chains
M <- 4 # number of chains

# Toy example with 3 i.i.d chains uniform in [-0.5, 0.5] and 1 in [-1, 1]:
chains <- array(c(runif((M-1)*N, -1/2, 1/2), runif(N, -1,1)), c(N,M))

# Quantile to evaluate:
x <- 0.5
# local_rhat(x, chains)
```

---

multivariate_all_local_rhat

Computation of multivariate $\hat{R}(x)$ on a given grid

Description

Return the vector of $\hat{R}(x)$ values that will be used to compute the supremum $\hat{R}_\infty$ in the multivariate case. See `multivariate_grid_for_R` for the grid computation for x.

Usage

```
multivariate_all_local_rhat(chains, dir = NULL, max_nb_points = 500)
```

Arguments

`chains`	an array of size $n \times d \times m$, where n is the length of the chains, $m \geq 2$ is the number of chains and d is the dimension.
`dir`	a binary vector of size d indicating the signs in the indicator variable. For example, `dir = c(0,...,0)` correspond to the computation of $\hat{R}$ on $I\{\theta_1^{(j)} \leq x_1, \ldots, \theta_d^{(j)} \leq x_d\}$. This direction will be used if no arguments is given.
`max_nb_points`	the maximal length of the grid in the case where the total number of samples is larger. By default, `max_nb_points = 500`.

Details

The computation of $\hat{R}_\infty$ require to compute a supremum of $\hat{R}(x)$ values over the quantiles x. See the documentation of `multivariate_grid_for_R` and `grid_for_R` to see how the choice of the different x is done. Thus, this function combine `multivariate_grid_for_R` for the computation of the grid and `multivariate_directed_local_rhat` or `multivariate_directed_local_rhat`, depending on if a direction is specified.

Value

A list that contains the different $\hat{R}(x)$ values used for the computation of $\hat{R}_\infty$.

References

TO INCLUDE

`multivariate_directed_local_rhat`

Multivariate $\hat{R}(x)$ with a specified dependence direction

---

Description

Compute $\hat{R}(x)$ in the multivariate case, for a given quantile x and a given sense for the d signs in the indicator variable.

Usage

```
multivariate_directed_local_rhat(x, chains, dir)
```

Arguments

`x`	a vector of size d corresponding to the quantile used for the computation of $\hat{R}(x)$.
`chains`	an array of size $n \times d \times m$, where n is the length of the chains, $m \geq 2$ is the number of chains and d is the dimension.
`dir`	a binary vector of size d indicating the signs in the indicator variable. For example, `dir = c(0,...,0)` correspond to the computation of $\hat{R}$ on $I\{\theta_1^{(j)} \leq x_1, \ldots, \theta_d^{(j)} \leq x_d\}$, and if `dir[i] == 1`, then the indicator with $\theta_d^{(j)} \geq x_d$ will be used.

Details

In the multivariate case, $\hat{R}$ can be computed on $I\{\theta_1^{(j)} \leq x_1, \ldots, \theta_d^{(j)} \leq x_d\}$, but also with indicator where some "$\leq$" are replaced by "$\geq$", which leads to 2^{d-1} possibilities and potentially as many different results of $\hat{R}_\infty$. Here, a 0-1 vector has to be mentioned in the function to specify which indicator to use, indicating which size on each dimension.

Value

The corresponding multivariate $\hat{R}(x)$ computed on the given dependence direction.

References

TO INCLUDE

Examples

```
library("localrhat")
library("MASS") # For Multivariate Normal Distribution
rho <- 0.9
m <- 4
n <- 100
reps <- 50

# Function to generate bivariate normal chains, with one that differs by its sigma:
gen_bvnormal_chains <- function(M, N, rho){
    sig_matrix <- (1-rho) * diag(2) + matrix(rho, nrow=2, ncol=2)
    return (array(c(mvrnorm((M-1)*N, mu = rep(0, 2), Sigma = diag(2)),
                    mvrnorm(N, mu = rep(0, 2), Sigma = sig_matrix)), c(N,2,M)))
}
chains <- gen_bvnormal_chains(m, n, rho)

# Quantile to evaluate:
x <- c(0.5, 0.5)
# multivariate_local_rhat(x, chains)
# multivariate_directed_local_rhat(x, chains)
```

multivariate_grid_for_R

Quantile values used for the computation of multivariate $\hat{R}\_\infty$

Description

Return the set of points that will be used to estimate the supremum of $\hat{R}(x)$ over $x \in R^d$. The function works in the same way as `grid_for_R` in the univariate case, see the corresponding documentation for more details.

Usage

```
multivariate_grid_for_R(chains, max_nb_points = 500)
```

Arguments

`chains`	an array of size $n \times d \times m$, where n is the length of the chains, $m \geq 2$ is the number of chains and d is the dimension.
`max_nb_points`	the maximal length of the grid in the case where the total number of samples is larger. By default, `max_nb_points` = `500`.

Value

A list that contains the different x used for the computation of $\hat{R}(x)$.

References

TO INCLUDE

multivariate_local_rhat

Multivariate version of the local-$\hat{R}$: $\hat{R}(x)$

Description

Compute $\hat{R}(x)$ in the multivariate case, which correspond to $\hat{R}$ computed on a multivariate indicator variable for a given quantile x.

Usage

```
multivariate_local_rhat(x, chains)
```

Arguments

`x`	a vector of size d corresponding to the quantile used for the computation of $\hat{R}(x)$.
`chains`	an array of size $n \times d \times m$, where n is the length of the chains, $m \geq 2$ is the number of chains and d is the dimension.

Details

The function return $\hat{R}$ computed on

$$I\{\theta_1^{(j)} \leq x_1, \ldots, \theta_d^{(j)} \leq x_d\}$$

, for a given value of $x \in R^d$.

Value

The multivariate $\hat{R}(x)$ computed on the m chains.

References

TO INCLUDE

Examples

```
library("localrhat")
library("MASS") # For Multivariate Normal Distribution
rho <- 0.9
m <- 4
n <- 100
reps <- 50
```

```
# Function to generate bivariate normal chains, with one that differs by its sigma:
gen_bvnormal_chains <- function(M, N, rho){
    sig_matrix <- (1-rho) * diag(2) + matrix(rho, nrow=2, ncol=2)
    return (array(c(mvrnorm((M-1)*N, mu = rep(0, 2), Sigma = diag(2)),
                    mvrnorm(N, mu = rep(0, 2), Sigma = sig_matrix)), c(N,2,M)))
}
chains <- gen_bvnormal_chains(m, n, rho)

# Quantile to evaluate:
x <- c(0.5, 0.5)
# multivariate_local_rhat(x, chains)
```

rhat_infinity *Scalar version of the local Gelman Rubin diagnostic $\hat{R}(x)$.*

Description

Compute $\hat{R}_\infty$, a scalar summary of the function $\hat{R}(x)$ corresponding to the supremum over the quantiles x.

Usage

```
rhat_infinity(chains, dir = NULL, max_nb_points = 500)
```

Arguments

`chains` an array of size $n \times m$ in the univariate case and of size $n \times d \times m$ in the d-variate one, where n is the length of the chains, $m \geq 2$ is the number of chains and d is the dimension.

`dir` a vector specifying which indicator to use for the multivariate case. See the function `multivariate_directed_local_rhat` for more details. If no direction is given, the computation is done on the indicator variable with the "$\leq$" sign on all dimension.

`max_nb_points` the maximal length of the grid in the case where the total number of samples is larger. By default, `max_nb_points = 500`.

Details

$\hat{R}_\infty$ is based on $\hat{R}(x)$, the computation of $\hat{R}$ on $I(\theta^{(i,j)} \leq x)$ (in the univariate case):

$$R_\infty = \sup_{x \in R} R(x).$$

This require to compute $\hat{R}(x)$ on different values of x to estimate this supremum. See the function `all_local_rhat` for more details. In the multivariate case, $\hat{R}$ can be computed on $I\{\theta_1^{(j)} \leq x_1, \dots, \theta_d^{(j)} \leq x_d\}$, but also with indicator where some "$\leq$ are replaced by "$\geq$", which leads to 2^{d-1} possibilities and potentially as many different results of $\hat{R}_\infty$. Here, a 0-1 vector can be given in the function to specify which indicator to use.

Value

The value of $\hat{R}_\infty$ computed on the m chains.

References

TO INCLUDE

Examples

```
library(localrhat)
library("MASS") # For Multivariate Normal Distribution
rho <- 0.9
m <- 4
n <- 100
reps <- 50

# Function to generate bivariate normal chains, with one that differs by its sigma:
gen_bvnormal_chains <- function(M, N, rho){
    sig_matrix <- (1-rho) * diag(2) + matrix(rho, nrow=2, ncol=2)
    return (array(c(mvrnorm((M-1)*N, mu = rep(0, 2), Sigma = diag(2)),
                    mvrnorm(N, mu = rep(0, 2), Sigma = sig_matrix)), c(N,2,M)))
}

R_values <- c()
for (e in 1:reps){
    chains <- gen_bvnormal_chains(m, n, rho)
    # R_values <- c(R_values, rhat_infinity_max_directions(chains))
}
# R_mat <- matrix(data = R_values, ncol = 1)
# colnames(R_mat) <- "R-hat-inf_max_dir"

# plot_hist(R_mat, bin_size = 0.004,
#           lim_y_axis = 15, vaxis_pos = 1.015,
#           plot_threshold = F)
```

`rhat_infinity_max_directions`

Symmetrical version of $\hat{R}$_∞ in the multivariate case.

Description

Compute the multivariate version of $\hat{R}_\infty$ in all possible dependence direction and return the maximum values.

Usage

```
rhat_infinity_max_directions(chains, max_nb_points = 500, nb_directions = NULL)
```

Arguments

`chains`	an array of size $n \times d \times m$, where n is the length of the chains, $m \geq 2$ is the number of chains and d is the dimension.
`max_nb_points`	the maximal length of the grid in the case where the total number of samples is larger. By default, `max_nb_points = 500`.
`nb_directions`	an integer giving the number of directions to compute $\hat{R}_\infty$: if given, it has to be between 1 and 2^{d-1} and a a random set of directions of this size will be used.

Details

In the multivariate case, the sensitivity of R_∞ strongly depends on the sign of dependence of the variables. This function implement the naive way to consider all direction, by computing on the 2^{d-1} possible one. **Warning**: this method becomes extremely costly when the dimension grow, and we recommend to avoid it as soon as $d > 5$ (pending future work on multidimensional improvements). Alternatively, a number of directions can be specified, to avoid computational issues and compute R_∞ on a subset of the possibilities: then, random directions will be generated to diagnostic the convergence on the corresponding dependence direction.

Value

The maximum value of all $\hat{R}_\infty$ computed.

Examples

```
library(localrhat)
library("MASS") # For Multivariate Normal Distribution
rho <- 0.9
m <- 4
n <- 100
reps <- 50

# Function to generate bivariate normal chains, with one that differs by its sigma:
gen_bvnormal_chains <- function(M, N, rho){
    sig_matrix <- (1-rho) * diag(2) + matrix(rho, nrow=2, ncol=2)
    return (array(c(mvrnorm((M-1)*N, mu = rep(0, 2), Sigma = diag(2)),
                    mvrnorm(N, mu = rep(0, 2), Sigma = sig_matrix)), c(N,2,M)))
}

R_values <- c()
for (e in 1:reps){
    chains <- gen_bvnormal_chains(m, n, rho)
    # R_values <- c(R_values, rhat_infinity_max_directions(chains))
}
# R_mat <- matrix(data = R_values, ncol = 1)
# colnames(R_mat) <- "R-hat-inf_max_dir"

# plot_hist(R_mat, bin_size = 0.005,
#           lim_y_axis = 15, vaxis_pos = 1.025,
#           plot_threshold = F)
```

`trad_rhat` *Compute the traditional Gelman-Rubin diagnostic.*

Description

Compute the traditional Gelman-Rubin diagnostic.

Usage

```
trad_rhat(chains)
```

Arguments

`chains`	an array of size $n \times m$ where n is the length of the chains and $m \geq 2$ is the number of chains.

Details

The function return the tradition Gelman-Rubin diagnostic $\hat{R}$.

Value

$\hat{R}$ computed on the m chains.

References

TO INCLUDE

Index