Package: hdbcp (via r-universe)

October 25, 2026

Type Package

Title Bayesian Change Point Detection for High-Dimensional Data

Version 1.0.0

Maintainer JaeHoon Kim <jhkimstat@gmail.com>

Description Functions implementing change point detection methods using the maximum pairwise Bayes factor approach. Additionally, the package includes tools for generating simulated datasets for comparing and evaluating change point detection techniques.

License GPL-3

Encoding UTF-8

URL https://github.com/JaeHoonKim98/hdbcp

BugReports https://github.com/JaeHoonKim98/hdbcp/issues

RoxygenNote 7.3.2

Imports Rcpp, stats, dplyr

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Author JaeHoon Kim [aut, cre], KyoungJae Lee [aut, ths]

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

Date/Publication 2024-11-19 03:50:02 UTC

RemoteUrl https://github.com/cran/hdbcp

RemoteRef HEAD

RemoteSha 11106cfd1093603220c25f75455fa458438f2390

Contents

generate_cov_datasets 2
generate_mean_datasets 3
majority_rule_mxPBF 4
mvrnorm_cpp 5

mxPBF_combined 6
mxPBF_cov 7
mxPBF_mean 9

Index **11**

`generate_cov_datasets` *Generate Simulated Datasets with Change Points in Covariance Matrix*

Description

This function generates simulated datasets that include change points in the covariance matrix for change point detection. Users can specify various parameters to control the dataset size, dimension, size of signal, and change point locations. The generated datasets include datasets with and without change points, allowing for comparisons in simulation studies.

Usage

```
generate_cov_datasets(
  n,
  p,
  signal_size,
  sparse = TRUE,
  single_point = round(n/2),
  multiple_points = c(round(n/4), round(2 * n/4), round(3 * n/4)),
  type = c(1, 2, 3, 4, 5)
)
```

Arguments

`n`	Number of observations to generate.
`p`	Number of features or dimensions for each observation.
`signal_size`	Magnitude of the signal applied at change points.
`sparse`	Determines if a sparse covariance structure is used (default is TRUE).
`single_point`	Location of a single change point in the dataset (default is n/2).
`multiple_points`	Locations of multiple change points within the dataset (default is quartiles of n).
`type`	Integer vector specifying the type of dataset to return. Options are as follows: - 1: No change points (H0 data) - 2: Single change point with rare signals - 3: Single change point with many signals - 4: Multiple change points with rare signals - 5: Multiple change points with many signals

Value

A 3D array containing the generated datasets. Each slice represents a different dataset type.

Examples

```
# Generate a default dataset
datasets <- generate_cov_datasets(100, 50, 1)

null_data <- datasets[,,1]
single_many_data <- datasets[,,3]
```

`generate_mean_datasets` *Generate Simulated Datasets with Change Points in Mean Vector*

Description

This function generates simulated datasets that include change points in the mean vector for change point detection. Users can specify various parameters to control the dataset size, dimension, size of signal, and change point locations. The generated datasets include datasets with and without change points, allowing for comparisons in simulation studies.

Usage

```
generate_mean_datasets(
  n = 500,
  p = 200,
  signal_size = 1,
  pre_proportion = 0.4,
  pre_value = 0.3,
  single_point = round(n/2),
  multiple_points = c(round(n/4), round(2 * n/4), round(3 * n/4)),
  type = c(1, 2, 3, 4, 5)
)
```

Arguments

`n`	Number of observations to generate.
`p`	Number of features or dimensions for each observation.
`signal_size`	Magnitude of the signal to apply at change points.
`pre_proportion`	Proportion of the covariance matrix's off-diagonal elements to be set to a predefined value (default is 0.4).
`pre_value`	Value assigned to selected off-diagonal elements of the covariance matrix (default is 0.3).
`single_point`	Location of a single change point in the dataset (default is n/2).
`multiple_points`	Locations of multiple change points within the dataset (default is quartiles of n).

type — Integer specifying the type of dataset to return. Options are as follows: - 1: No change points (H0 data) - 2: Single change point with rare signals - 3: Single change point with many signals - 4: Multiple change points with rare signals - 5: Multiple change points with many signals The default options are 1, 2, 3, 4, and 5.

Value

A 3D array containing the generated datasets. Each slice represents a different dataset type.

Examples

```
# Generate a default dataset
datasets <- generate_mean_datasets(100, 50, 1)

null_data <- datasets[,,1]
single_many_data <- datasets[,,3]
```

---

majority_rule_mxPBF *Majority Rule for Multiscale approach using mxPBF Results*

Description

This function implements a majority rule-based post-processing approach to identify common change points across multiple window sizes from mxPBF results.

Usage

```
majority_rule_mxPBF(res_mxPBF)
```

Arguments

res_mxPBF — A list of results from `mxPBF_mean()` or `mxPBF_cov()`.

Value

A vector of final detected change points that are common across multiple windows based on majority rule.

Examples

```
n <- 500
p <- 200
signal_size <- 1
pre_value <- 0.3
pre_proportion <- 0.4
given_data <- generate_mean_datasets(n, p, signal_size, pre_proportion, pre_value,
single_point = 250, multiple_points = c(150,300,350), type = 5)
```

```
nws <- c(25, 60, 100)
alps <- seq(1,10,0.05)
res_mxPBF <- mxPBF_mean(given_data, nws, alps)
majority_rule_mxPBF(res_mxPBF)
```

`mvrnorm_cpp` *Multivariate Normal Random Number Generator*

Description

Generates random numbers from a multivariate normal distribution with specified mean and covariance matrix using a C++ implementation.

Usage

```
mvrnorm_cpp(n = 1, mu, Sigma)
```

Arguments

`n`	The number of random samples to generate. Defaults to 1.
`mu`	The mean vector of the distribution.
`Sigma`	The covariance matrix of the distribution.

Value

A numeric matrix where each row is a random sample from the multivariate normal distribution.

Examples

```
# Example usage
mu <- c(0, 0)
Sigma <- matrix(c(1, 0.5, 0.5, 1), 2, 2)
mvrnorm_cpp(5, mu, Sigma)
```

`mxPBF_combined` *Change Point Detection in Mean Structure using Maximum Pairwise Bayes Factor (mxPBF)*

Description

This function detects change points in both mean and covariance structure of multivariate Gaussian data using the Maximum Pairwise Bayes Factor (mxPBF). The function selects alpha that controls the empirical False Positive Rate (FPR), as suggested in the paper. The function conducts a multi-scale approach using the function.

Usage

```
mxPBF_combined(
  given_data,
  nws,
  alps,
  a0 = 0.01,
  b0 = 0.01,
  FPR_want = 0.05,
  n_sample = 300,
  n_cores = 1
)
```

Arguments

`given_data`	An $(n \times p)$ data matrix representing n observations and p variables.
`nws`	A set of window sizes for change point detection.
`alps`	A grid of alpha values used in the empirical False Positive Rate (FPR) method.
`a0`	A hyperparameter a_0 used in the mxPBF (default: 0.01).
`b0`	A hyperparameter b_0 used in the mxPBF (default: 0.01).
`FPR_want`	Desired False Positive Rate for selecting alpha, used in the empirical FPR method (default: 0.05).
`n_sample`	Number of simulated samples to estimate the empirical FPR, used in the empirical FPR method (default: 300).
`n_cores`	Number of threads for parallel execution via OpenMP (default: 1).

Value

A list provided. Each element in the list contains:

Result_cov A list result from the `mxPBF_cov()` function.

Result_mean A list result from the `mxPBF_mean()` function applied to each segmented data.

Change_points_cov Locations of detected change points identified by `mxPBF_cov()` function.

Change_points_mean Locations of detected change points identified by `mxPBF_mean()` function.

Examples

```
nws <- c(25, 60, 100)
alps <- seq(1,10,0.05)
## H0 data
mu1 <- rep(0,10)
sigma1 <- diag(10)
X <- mvrnorm_cpp(500, mu1, sigma1)
res1 <- mxPBF_combined(X, nws, alps)

## H1 data
mu2 <- rep(1,10)
sigma2 <- diag(10)
for (i in 1:10) {
  for (j in i:10) {
    if (i == j) {
    next
    } else {
    cov_value <- rnorm(1, 1, 1)
    sigma2[i, j] <- cov_value
    sigma2[j, i] <- cov_value
    }
  }
}
sigma2 <- sigma2 + (abs(min(eigen(sigma2)$value))+0.1)*diag(10) # Make it nonsingular
Y1 <- mvrnorm_cpp(150, mu1, sigma1)
Y2 <- mvrnorm_cpp(150, mu2, sigma1)
Y3 <- mvrnorm_cpp(200, mu2, sigma2)
Y <- rbind(Y1, Y2, Y3)
res2 <- mxPBF_combined(Y, nws, alps)
```

`mxPBF_cov` *Change Point Detection in Covaraiance Structure using Maximum Pairwise Bayes Factor (mxPBF)*

Description

This function detects change points in the covariance structure of multivariate Gaussian data using the Maximum Pairwise Bayes Factor (mxPBF). The function selects alpha that controls the empirical False Positive Rate (FPR), as suggested in the paper. One can conduct a multiscale approach using the function `majority_rule_mxPBF()`.

Usage

```
mxPBF_cov(
  given_data,
  nws,
  alps,
```

```
  a0 = 0.01,
  b0 = 0.01,
  FPR_want = 0.05,
  n_sample = 300,
  n_cores = 1
)
```

Arguments

given_data	An $(n \times p)$ data matrix representing n observations and p variables.
nws	A set of window sizes for change point detection.
alps	A grid of alpha values used in the empirical False Positive Rate (FPR) method.
a0	A hyperparameter a_0 used in the mxPBF (default: 0.01).
b0	A hyperparameter b_0 used in the mxPBF (default: 0.01).
FPR_want	Desired False Positive Rate for selecting alpha, used in the empirical FPR method (default: 0.05).
n_sample	Number of simulated samples to estimate the empirical FPR, used in the empirical FPR method (default: 300).
n_cores	Number of threads for parallel execution via OpenMP (default: 1).

Value

A list of length equal to the number of window sizes provided. Each element in the list contains:

Change_points Locations of detected change points.

Bayes_Factors Vector of calculated Bayes Factors for each middle points.

Selected_alpha Optimal alpha value selected based on the method that controls the empirical FPR.

Window_size Window size used for change point detection.

Examples

```
nws <- c(25, 60, 100)
alps <- seq(1,10,0.05)
## H0 data
mu <- rep(0,10)
sigma1 <- diag(10)
X <- mvrnorm_cpp(500, mu, sigma1)
res1 <- mxPBF_cov(X, nws, alps)

## H1 data
mu <- rep(0,10)
sigma2 <- diag(10)
for (i in 1:10) {
  for (j in i:10) {
    if (i == j) {
    next
    } else {
    cov_value <- rnorm(1, 1, 1)
```

```
    sigma2[i, j] <- cov_value
    sigma2[j, i] <- cov_value
    }
  }
}
sigma2 <- sigma2 + (abs(min(eigen(sigma2)$value))+0.1)*diag(10) # Make it nonsingular
Y1 <- mvrnorm_cpp(250, mu, sigma1)
Y2 <- mvrnorm_cpp(250, mu, sigma2)
Y <- rbind(Y1, Y2)
res2 <- mxPBF_cov(Y, nws, alps)
```

`mxPBF_mean` *Change Point Detection in Mean Structure using Maximum Pairwise Bayes Factor (mxPBF)*

Description

This function detects change points in the mean structure of multivariate Gaussian data using the Maximum Pairwise Bayes Factor (mxPBF). The function selects alpha that controls the empirical False Positive Rate (FPR), as suggested in the paper. One can conduct a multiscale approach using the function `majority_rule_mxPBF()`.

Usage

```
mxPBF_mean(given_data, nws, alps, FPR_want = 0.05, n_sample = 300, n_cores = 1)
```

Arguments

`given_data`	An $(n \times p)$ data matrix representing n observations and p variables.
`nws`	A set of window sizes for change point detection.
`alps`	A grid of alpha values used in the empirical False Positive Rate (FPR) method.
`FPR_want`	Desired False Positive Rate for selecting alpha, used in the empirical FPR method (default: 0.05).
`n_sample`	Number of simulated samples to estimate the empirical FPR, used in the empirical FPR method (default: 300).
`n_cores`	Number of threads for parallel execution via OpenMP (default: 1).

Value

A list of length equal to the number of window sizes provided. Each element in the list contains:

Change_points Locations of detected change points.

Bayes_Factors Vector of calculated Bayes Factors for each middle points.

Selected_alpha Optimal alpha value selected based on the method that controls the empirical FPR.

Window_size Window size used for change point detection.

Examples

```
nws <- c(25, 60, 100)
alps <- seq(1,10,0.05)
## H0 data
mu1 <- rep(0,10)
sigma <- diag(10)
X <- mvrnorm_cpp(500, mu1, sigma)
res1 <- mxPBF_mean(X, nws, alps)

## H1 data
mu2 <- rep(1,10)
sigma <- diag(10)
Y <- rbind(mvrnorm_cpp(250,mu1,sigma), mvrnorm_cpp(250,mu2,sigma))
res2 <- mxPBF_mean(Y, nws, alps)
```

Index

generate_cov_datasets, 2
generate_mean_datasets, 3

majority_rule_mxPBF, 4
mvrnorm_cpp, 5
mxPBF_combined, 6
mxPBF_cov, 7
mxPBF_mean, 9