Package: roboBayes (via r-universe)

October 26, 2026

Type Package

Title Robust Online Bayesian Monitoring

Version 1.3

Date 2025-05-04

Author Laura Wendelberger [aut], Josh Gray [aut], Brian Reich [aut], Alyson Wilson [aut], Shannon T. Holloway [aut, cre]

Maintainer Shannon T. Holloway <shannon.t.holloway@gmail.com>

Description An implementation of Bayesian online changepoint detection (Adams and MacKay (2007) <doi:10.48550/arXiv.0710.3742>) with an option for probability based outlier detection and removal (Wendelberger et. al. (2021) <doi:10.48550/arXiv.2112.12899>). Building on the independent multivariate constant mean model implemented in the 'R' package 'ocp', this package models multivariate data as multivariate normal about a linear trend, defined by user input covariates, with an unstructured error covariance. Changepoints are identified based on a probability threshold for windows of points.

License GPL-2

Depends R (>= 3.5.0), methods

Imports Rcpp (>= 1.0.7), RcppArmadillo

Suggests mvtnorm

NeedsCompilation yes

LinkingTo Rcpp, RcppArmadillo, RcppDist

Encoding UTF-8

RoxygenNote 7.2.3

Collate 'RcppExports.R' 'checkTypes.R' 'regressionInit.R' 'initRoboBayes.R' 'verifyAnalysisSettings.R' 'roboBayes.R'

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

Date/Publication 2025-05-04 18:40:39 UTC

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

RemoteRef HEAD

RemoteSha 97d26d5104e34ec9fa537658b6a9d613fb3b9d81

Contents

roboBayes 2

Index **7**

---

roboBayes *roboBayes: Robust Online Bayesian Monitoring*

Description

The primary function for Robust Online Bayesian Monitoring (roboBayes). Performs roboBayes on the response data regressed on the input covariates, which should account for nonstationarity in the response between changepoints. The algorithm outputs changepoints based on a threshold, a roboBayes object incorporating data thus far, and algorithm settings. A roboBayes object can be updated with new data by including it in a successive roboBayes() function call.

Usage

```
roboBayes(
  datapts,
  covariates,
  RoboBayes = NULL,
  ...,
  lambda = 1000,
  par_inits = NULL,
  truncRthresh = 1e-04,
  truncRmin = 100L,
  cpthresh = 0.8,
  cptimemin = 4L,
  Lgroup = 3L,
  Lsearch = 10L,
  Lwindow = 30L,
  Lm = 7L,
  alpha = 0.9,
  kt = 0L,
  pc = 0.5,
  cp_delay = 3L,
  outlier_mean = rep(x = 0.5, times = ncol(x = datapts)),
  outlier_var = diag(x = 2, nrow = ncol(x = datapts)),
  getR = FALSE,
  getOutliers = TRUE,
  getModels = FALSE
)
```

Arguments

datapts	A matrix object (n x d). The observations for n time points. There are methods in place to attempt to convert non-matrix input (data.frame, numeric) to a matrix. If a numeric vector is provided, it is assumed that d = 1.
covariates	A matrix object (n x k). The covariates for n time points. If any covariates depend on the run length (kt > 0), these covariates must be provided in the last kt columns of the input matrix. There are methods in place to attempt to convert non-matrix input (data.frame, numeric) to a matrix. If a numeric vector is provided, it is assumed that k = 1.
RoboBayes	A value object returned by a prior roboBayes() analysis or NULL. If NULL, the analysis is taken to be the first step, and all analysis settings (inputs after the ellipsis) can be adjusted. For all subsequent steps, RoboBayes is the value object of the preceding step and analysis settings are taken from that object – any analysis setting inputs provided for continuation steps are ignored.
...	Ignored. Included only to require named inputs. If RoboBayes is an object of class RoboBayes, all inputs after the ellipsis are ignored.
lambda	A numeric object. The prior probability of a changepoint occurring at any time point.
par_inits	A list object. The initial estimates for the hyperparameters. The list can contain one or more of the following: • B: A (k x d) matrix. The location parameter of the matrix Normal. The default is a zero matrix. • V: A (d x d) matrix. The scale matrix of the Inverse Wishart. The default is a symmetric matrix with 1.0 on the diagonal and 0.1 in the off-diagonal elements. • nu: A scalar. The degrees of freedom of the Inverse Wishart. The default is (d - 0.9). • Lambda: A (k x k) matrix. The scale parameter of the matrix Normal. The default is a diagonal matrix of 0.01.
truncRthresh	A scalar object. A probability threshold used to limit the size of the data used at each time step. Once t > truncRmin, run lengths associated with data points from times less than t-truncRmin for which the probability < truncRthresh are removed
truncRmin	An integer object. The minimum number of time points maintained before data truncation methods are utilized.
cpthresh	A numeric vector object. One or more changepoints.
cptimemin	An integer object. Minimum time at which updated change points can be ascertained.
Lgroup	An integer object. The size of the moving window used to aggregate run length probabilities during the initial coarse search for the presence of a changepoint.
Lsearch	An integer object. The number of run length probabilities to aggregate for calculating the probability of "recent" change.
Lwindow	An integer object. The number of points in the window to scan for the most recent changepoint. Changes further back than Lwindow + cp_delay will not be recorded.

Lm	An integer object. The number of data points to store in memory for outlier elimination. A change recorded more than Lm points previously will not be examined for being an outlier. When getModels is TRUE, Lm is automatically set to Lwindow + cp_delay + 1 to ensure that model parameters associated with detected changepoints can be recovered.
alpha	A scalar object. The probability threshold for declaring a point an outlier.
kt	An integer object. The number of covariates that depend on run length.
pc	A scalar object. The prior probability of the model without an outlier being the correct model, given that a candidate change/outlier has been detected.
cp_delay	An integer object. The minimum number of points that must be observed after the change to declare a changepoint.
outlier_mean	A numeric vector (d). The mean value for the distribution of outlier values.
outlier_var	A matrix object (d x d). The covariance matrix for the distribution of outlier values.
getR	A logical object. If TRUE, the matrix of probabilities is returned. Warning – selecting TRUE will increase computation time, and for large analyses, may exceed available memory.
getOutliers	A logical object. Detect outliers. If TRUE, the matrix of probabilities used for outlier elimination are returned. Selecting TRUE will increase computation time.
getModels	A logical object. If TRUE, the model parameters of each changepoint are returned, as well as the matrix of probabilities used for outlier elimination. When TRUE, parameter Lm is set to Lwindow + cp_delay + 1, regardless of that provided as input.

Details

All inputs after the ellipsis are analysis settings that can only be set in the initial step of the roboBayes procedure. In the first step, input roboBayes must be NULL. All subsequent steps of the procedure must provide, through input roboBayes, the value object returned by the previous step.

Analysis parameters must obey the following conditions:

- Lsearch + cp_delay + Lgroup <= truncRmin + 1
- Lwindow + cp_delay + Lgroup <= truncRmin + 1
- Lgroup <= Lwindow + 1
- cptimemin > Lgroup
- cptimemin >= cp_delay - 1
- cp_delay < Lm
- Lsearch <= truncRmin

Value

A RoboBayes object, which extends list and contains

- R: A numeric vector object. The current posterior distribution of the run length.
- RL: An integer vector object. The current run lengths that are retained.
- truncRind: An integer vector object. The indices of the run lengths retained.
- jtR: A numeric vector object. The current joint distribution of the run length and data.
- pars: A list object. Elements are run length specific summary statistics and hyperparameters. With the exception of nu, each hyperparameter/summary statistic is a 3 dimensional array, where the final dimension corresponds to the run length. The degrees of freedom, nu, are returned as a vector, of length equal to the number of run lengths retained.
- cpInds: An integer matrix object (length(cpthresh) x nRuns). Each element contains the most recent changepoint.
- lastLs: A numeric vector object. Each element contains the probability that a change occurred in the previous Lsearch time points, delayed by cp_delay points.
- time: An integer object. The current time.
- allcov: A numeric matrix object. The run length dependent covariates for the retained run lengths.
- model0: A list object. The initial hyperparameters.
- lastDataPt: A numeric vector object. The data of the last time point.
- call: The matched call.
- params: A list object. The analysis settings.

Conditionally returned elements include:

If getR = TRUE

- RFull: A numeric matrix object. The ith column contains the run length posterior distribution for time i.

If getOutliers = TRUE

- Rm: A numeric matrix object. The ith column contains the joint distribution of data and run length associated with time-i+1.
- y_store: A numeric matrix object. Contains the most recent Lm data points.
- x_store: A numeric matrix object. Contains the most recent Lm covariates.
- outliers: An integer vector object. Time points that have been identified as outliers.

If getModels = TRUE

- Rm: A numeric matrix object. The ith column contains the joint distribution of data and run length associated with time-i+1.
- y_store: A numeric matrix object. Contains the most recent Lm data points.
- x_store: A numeric matrix object. Contains the most recent Lm covariates.
- mods: A list object. Each element of the list corresponds to an identified changepoint. For each changepoint, the expected model coefficients and covariance.
- currentModel: A list object. For the current most likely run length, the expected model coefficients and covariance.

Examples

```
nt <- 100

## 2 covariates each time step

x <- cbind(rep(1,nt), rnorm(nt))

## 2 observations each time step

# covariance matrix
sigma <- matrix(data = -0.3, nrow = 2, ncol = 2)
diag(sigma) <- 1.0

# mean
mean1 <- c(0.5, 0.1)
mean2 <- c(0.2, 0.2)

y <- rbind(mvtnorm::rmvnorm(n = nt/2, mean = mean1, sigma = 0.001*sigma),
           mvtnorm::rmvnorm(n = nt/2, mean = mean2, sigma = 0.001*sigma))

# add in an outlier
y[30,] <- c(0.9,0.01)
y[y < 0.0] <- 0.0

step1 <- roboBayes(datapts = y[1:32,], covariates = x[1:32,])

step2 <- roboBayes(datapts = y[33:64,], covariates = x[33:64,], RoboBayes = step1)

step3 <- roboBayes(datapts = y[65:94,], covariates = x[65:94,], RoboBayes = step2)

step4 <- roboBayes(datapts = y[95:100,], covariates = x[95:100,], RoboBayes = step3)
```

Index

roboBayes, 2