Package 'eNchange'

May 8, 2026

Type Package

Title Ensemble Methods for Multiple Change-Point Detection

Version 1.1

Date 2025-06-22

Maintainer Karolos K. Korkas <kkorkas@yahoo.co.uk>

Description Implements a segmentation algorithm for multiple change-point detection in univariate time series using the Ensemble Binary Segmentation of Korkas (2022) <Journal of the Korean Statistical Society, 51(1), pp.65-86.>.

License GPL (>= 2)

Imports Rcpp (>= 0.12.12), foreach, iterators, doParallel, methods, hawkes, ACDm

Suggests MASS

LinkingTo Rcpp

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2025-06-24 23:50:02 UTC

Author Karolos K. Korkas [aut, cre]

Contents

---

eNchange-package *Ensemble Methods for Multiple Change-Point Detection*

Description

Implements a segmentation algorithm for multiple change-point detection in univariate time series using the Ensemble Binary Segmentation of Korkas (2022).

Details

We propose a new technique for consistent estimation of the number and locations of the change-points in the structure of an irregularly spaced time series. The core of the segmentation procedure is the Ensemble Binary Segmentation method (EBS), a technique in which a large number of multiple change-point detection tasks using the Binary Segmentation (BS) method are applied on sub-samples of the data of differing lengths, and then the results are combined to create an overall answer. This methodology is applied to irregularly time series models such as the time-varying Autoregressive Conditional Duration model or the time-varying Hawkes process.

Author(s)

Karolos K. Korkas <kkorkas@yahoo.co.uk>.

Maintainer: Karolos K. Korkas <kkorkas@yahoo.co.uk>

References

Korkas, K.K., 2022. Ensemble binary segmentation for irregularly spaced data with change-points. Journal of the Korean Statistical Society, 51(1), pp.65-86.

Examples

```
 ## Not run:
pw.acd.obj <- new("simACD")
pw.acd.obj@cp.loc <- seq(0.1,0.95,by=0.025)
pw.acd.obj@lambda_0 <- rep(c(0.5,2),1+length(pw.acd.obj@cp.loc)/2)
pw.acd.obj@alpha <- rep(0.2,1+length(pw.acd.obj@cp.loc))
pw.acd.obj@beta <- rep(0.4,1+length(pw.acd.obj@cp.loc))
pw.acd.obj@N <- 5000
pw.acd.obj <- pc_acdsim(pw.acd.obj)
ts.plot(pw.acd.obj@x,main="Ensemble BS");abline(v=EnBinSeg(pw.acd.obj@x)[[1]],col="red")
#real change-points in grey
abline(v=floor(pw.acd.obj@cp.loc*pw.acd.obj@N),col="grey",lty=2)
ts.plot(pw.acd.obj@x,main="Standard BS");abline(v=BinSeg(pw.acd.obj@x)[[1]],col="blue")
#real change-points in grey
abline(v=floor(pw.acd.obj@cp.loc*pw.acd.obj@N),col="grey",lty=2)


## End(Not run)
```

BinSeg *An S4 method to detect the change-points in an irregularly spaced time series using Binary Segmentation.*

Description

An S4 method to detect the change-points in an irregularly spaced time series using the Binary Segmentation methodology described in Korkas (2022).

Usage

```
BinSeg(
  H,
  thresh = "universal",
  q = 0.99,
  p = 1,
  z = NULL,
  start.values = c(0.9, 0.6),
  dampen.factor = "auto",
  epsilon = 1e-05,
  LOG = TRUE,
  process = "acd",
  acd_p = 0,
  acd_q = 1,
  do.parallel = 2
)

## S4 method for signature 'ANY'
BinSeg(
  H,
  thresh = "universal",
  q = 0.99,
  p = 1,
  z = NULL,
  start.values = c(0.9, 0.6),
  dampen.factor = "auto",
  epsilon = 1e-05,
  LOG = TRUE,
  process = "acd",
  acd_p = 0,
  acd_q = 1,
  do.parallel = 2
)
```

Arguments

H The input irregular time series.

thresh	The threshold parameter which acts as a stopping rule to detect further change-points and has the form C log(sample). If "universal" then C is data-independent and preselected using the approach described in Korkas (2022). If "boot" it uses the data-dependent method boot_thresh. Default is "universal".
q	The universal threshold simulation quantile or the bootstrap distribution quantile. Default is 0.99.
p	The support of the CUSUM statistic. Default is 1.
z	Transform the time series to use for post-processing. If NULL this is done automatically. Default is NULL.
start.values	Warm starts for the optimizers of the likelihood functions.
dampen.factor	The dampen factor in the denominator of the residual process. Default is "auto".
epsilon	A parameter added to ensure the boundness of the residual process. Default is 1e-5.
LOG	Take the log of the residual process. Default is TRUE.
process	Choose between acd or hawkes. Default is acd.
acd_p	The p order of the ACD model. Default is 0.
acd_q	The q order of the ACD model. Default is 1.
do.parallel	Choose the number of cores for parallel computation. If 0 no parallelism is done. Default is 2. (Only applies if thresh = "boot").

Value

Returns a list with the detected change-points and the transformed series.

References

Korkas, K.K., 2022. Ensemble binary segmentation for irregularly spaced data with change-points. Journal of the Korean Statistical Society, 51(1), pp.65-86.

Examples

```
pw.acd.obj <- new("simACD")
pw.acd.obj@cp.loc <- seq(0.1,0.95,by=0.025)
pw.acd.obj@lambda_0 <- rep(c(0.5,2),1+length(pw.acd.obj@cp.loc)/2)
pw.acd.obj@alpha <- rep(0.2,1+length(pw.acd.obj@cp.loc))
pw.acd.obj@beta <- rep(0.4,1+length(pw.acd.obj@cp.loc))
pw.acd.obj@N <- 5000
pw.acd.obj <- pc_acdsim(pw.acd.obj)
ts.plot(pw.acd.obj@x,main="Standard BS");abline(v=BinSeg(pw.acd.obj@x)[[1]],col="blue")
#real change-points in grey
abline(v=floor(pw.acd.obj@cp.loc*pw.acd.obj@N),col="grey",lty=2)
```

boot_thresh *A bootstrap method to calculate the threshold (stopping rule) in the BS or EBS segmentation.*

Description

A bootstrap method to calculate the threshold (stopping rule) in the BS or EBS segmentation described in Cho and Korkas (2022) and adapted for irregularly time series in Korkas (2022).

Usage

```
boot_thresh(
  H,
  q = 0.75,
  r = 100,
  p = 1,
  start.values = c(0.9, 0.6),
  process = "acd",
  do.parallel = 2,
  dampen.factor = "auto",
  epsilon = 1e-05,
  LOG = TRUE,
  acd_p = 0,
  acd_q = 1
)

## S4 method for signature 'ANY'
boot_thresh(
  H,
  q = 0.75,
  r = 100,
  p = 1,
  start.values = c(0.9, 0.6),
  process = "acd",
  do.parallel = 2,
  dampen.factor = "auto",
  epsilon = 1e-05,
  LOG = TRUE,
  acd_p = 0,
  acd_q = 1
)
```

Arguments

H	The input irregular time series.
q	The bootstrap distribution quantile. Default is 0.75.

r	The number of bootrstap simulations. Default is 100.
p	The support of the CUSUM statistic. Default is 1.
start.values	Warm starts for the optimizers of the likelihood functions.
process	Choose between acd or hawkes. Default is acd.
do.parallel	Choose the number of cores for parallel computation. If 0 no parallelism is done. Default is 2.
dampen.factor	The dampen factor in the denominator of the residual process. Default is "auto".
epsilon	A parameter added to ensure the boundness of the residual process. Default is 1e-5.
LOG	Take the log of the residual process. Default is TRUE.
acd_p	The p order of the ACD model. Default is 0.
acd_q	The q order of the ACD model. Default is 1.

Value

Returns the threshold C.

References

Cho, H. and Korkas, K.K., 2022. High-dimensional GARCH process segmentation with an application to Value-at-Risk. Econometrics and Statistics, 23, pp.187-203.

Examples

```
pw.acd.obj <- new("simACD")
pw.acd.obj@cp.loc <- c(0.25,0.75)
pw.acd.obj@lambda_0 <- c(1,2,1)
pw.acd.obj@alpha <- rep(0.2,3)
pw.acd.obj@beta <- rep(0.7,3)
pw.acd.obj@N <- 3000
pw.acd.obj <- pc_acdsim(pw.acd.obj)
boot_thresh(pw.acd.obj@x,r=20)
```

EnBinSeg *An S4 method to detect the change-points in an irregularly spaced time series using Ensemble Binary Segmentation.*

Description

An S4 method to detect the change-points in an irregularly spaced time series using the Ensemble Binary Segmentation methodology described in Korkas (2022).

Usage

```
EnBinSeg(
  H,
  thresh = "universal",
  q = 0.99,
  p = 1,
  start.values = c(0.9, 0.6),
  dampen.factor = "auto",
  epsilon = 1e-05,
  LOG = TRUE,
  process = "acd",
  thresh2 = 0.05,
  num_ens = 500,
  min_dist = 0.005,
  pp = 1,
  do.parallel = 2,
  b = NULL,
  acd_p = 0,
  acd_q = 1
)

## S4 method for signature 'ANY'
EnBinSeg(
  H,
  thresh = "universal",
  q = 0.99,
  p = 1,
  start.values = c(0.9, 0.6),
  dampen.factor = "auto",
  epsilon = 1e-05,
  LOG = TRUE,
  process = "acd",
  thresh2 = 0.05,
  num_ens = 500,
  min_dist = 0.005,
  pp = 1,
  do.parallel = 2,
  b = NULL,
  acd_p = 0,
  acd_q = 1
)
```

Arguments

H	The input irregular time series.
thresh	The threshold parameter which acts as a stopping rule to detect further change-points and has the form C log(sample). If "universal" then C is data-independent and preselected using the approach described in Korkas (2020). If "boot" it uses

	the data-dependent method boot_thresh. Default is "universal".
q	The universal threshold simulation quantile or the bootstrap distribution quantile. Default is 0.99.
p	The support of the CUSUM statistic. Default is 1.
start.values	Warm starts for the optimizers of the likelihood functions.
dampen.factor	The dampen factor in the denominator of the residual process. Default is "auto".
epsilon	A parameter added to ensure the boundness of the residual process. Default is 1e-5.
LOG	Take the log of the residual process. Default is TRUE.
process	Choose between "acd" or "hawkes" or "additive" (signal +iid noise). Default is "acd".
thresh2	Keep only the change-points that appear more than thresh2 M times.
num_ens	Number of ensembles denoted by M in the paper. Default is 500.
min_dist	The minimum distance as percentage of sample size to use in the post-processing. Default is 0.005.
pp	Post-process the change-points based on the distance from the highest ranked change-points.
do.parallel	Choose the number of cores for parallel computation. If 0 no parallelism is done. Default is 2.
b	A parameter to control how close the random end points are to the start points. A large value will on average return shorter random intervals. If NULL all points have an equal chance to be selected (uniformly distributed). Default is NULL.
acd_p	The p order of the ACD model. Default is 0.
acd_q	The q order of the ACD model. Default is 1.

Value

Returns a list with the detected change-points and the frequency table of the ensembles across M applications.

References

Korkas, K.K., 2022. Ensemble binary segmentation for irregularly spaced data with change-points. Journal of the Korean Statistical Society, 51(1), pp.65-86.

Examples

```
pw.acd.obj <- new("simACD")
pw.acd.obj@cp.loc <- seq(0.1,0.95,by=0.025)
pw.acd.obj@lambda_0 <- rep(c(0.5,2),1+length(pw.acd.obj@cp.loc)/2)
pw.acd.obj@alpha <- rep(0.2,1+length(pw.acd.obj@cp.loc))
pw.acd.obj@beta <- rep(0.4,1+length(pw.acd.obj@cp.loc))
pw.acd.obj@N <- 5000
pw.acd.obj <- pc_acdsim(pw.acd.obj)
ts.plot(pw.acd.obj@x,main="Ensemble BS");abline(v=EnBinSeg(pw.acd.obj@x)[[1]],col="red")
```

```
#real change-points in grey
abline(v=floor(pw.acd.obj@cp.loc*pw.acd.obj@N),col="grey",lty=2)
ts.plot(pw.acd.obj@x,main="Standard BS");abline(v=BinSeg(pw.acd.obj@x)[[1]],col="blue")
#real change-points in grey
abline(v=floor(pw.acd.obj@cp.loc*pw.acd.obj@N),col="grey",lty=2)
```

---

pc_acdsim-class *A method to simulate nonstationary ACD models.*

Description

A S4 method that takes as an input a simACD object and outputs a simulated nonstationary ACD(1,1) model. The formulation of the of the piecewise constant ACD model is given in the simACD class.

Usage

```
pc_acdsim(object)

## S4 method for signature 'ANY'
pc_acdsim(object)
```

Arguments

object	a simACD object

Value

Returns an object of simACD class containing a simulated piecewise constant ACD time series.

References

Korkas, K.K., 2022. Ensemble binary segmentation for irregularly spaced data with change-points. Journal of the Korean Statistical Society, 51(1), pp.65-86.

Examples

```
pw.acd.obj <- new("simACD")
pw.acd.obj@cp.loc <- c(0.25,0.75)
pw.acd.obj@lambda_0 <- c(1,2,1)
pw.acd.obj@alpha <- rep(0.2,3)
pw.acd.obj@beta <- rep(0.7,3)
pw.acd.obj@N <- 3000
pw.acd.obj <- pc_acdsim(pw.acd.obj)
ts.plot(pw.acd.obj@x)
ts.plot(pw.acd.obj@psi)
```

pc_hawkessim-class *A method to simulate nonstationary Hawkes models.*

Description

A S4 method that takes as an input a simHawkes object and outputs a simulated nonstationary Hawkes model. The formulation of the of the piecewise constant ACD model is given in the simHawkes class.

Usage

```
pc_hawkessim(object)

## S4 method for signature 'ANY'
pc_hawkessim(object)
```

Arguments

object	a simHawkes object

Value

Returns an object of simHawkes class containing a simulated piecewise constant Hawkes series.

References

Korkas, K.K., 2022. Ensemble binary segmentation for irregularly spaced data with change-points. Journal of the Korean Statistical Society, 51(1), pp.65-86.

Examples

```
pw.hawk.obj <- new("simHawkes")
pw.hawk.obj@cp.loc <- c(0.5)
pw.hawk.obj@lambda_0 <- c(1,2)
pw.hawk.obj@alpha <- c(0.2,0.2)
pw.hawk.obj@beta <- c(0.7,0.7)
pw.hawk.obj@horizon <- 1000
pw.hawk.obj <- pc_hawkessim(pw.hawk.obj)
ts.plot(pw.hawk.obj@H)
ts.plot(pw.hawk.obj@cH)
```

simACD-class *An S4 class for a nonstationary ACD model.*

Description

A specification class to create an object of a simulated piecewise constant conditional duration model of order (1,1). $x_t/\psi_t = \varepsilon_t \ \sim \mathcal{G}(\theta_2)\ \psi_t = \omega(t) + \sum_{j=1}^{p} \alpha_j(t) x_{t-j} + \sum_{k=1}^{q} \beta_k(t)\psi_{t-k}$. where $\psi_t = \mathcal{E}[x_t|x_t, \ldots, x_1|\theta_1]$ is the conditional mean duration of the t-th event with parameter vector θ_1 and $\mathcal{G}(.)$ is a general distribution over $(0, +\infty)$ with mean equal to 1 and parameter vector θ_2. In this work we assume that $\varepsilon_t \ \sim \exp(1)$.

Value

Returns an object of `simACD` class.

Slots

`x` The durational time series.

`psi` The psi time series.

`N` Sample sze of the time series.

`cp.loc` The vector with the location of the changepoints. Takes values from 0 to 1 or NULL. Default is NULL.

`lambda_0` The vector of the parameters lambda_0 in the ACD series as in the above formula.

`alpha` The vector of the parameters alpha in the ACD series as in the above formula.

`beta` The vector of the parameters beta in the ACD series as in the above formula.

`BurnIn` The size of the burn-in sample. Note that this only applies at the first simulated segment. Default is 500.

References

Korkas, K.K., 2022. Ensemble binary segmentation for irregularly spaced data with change-points. Journal of the Korean Statistical Society, 51(1), pp.65-86.

Examples

```
pw.acd.obj <- new("simACD")
pw.acd.obj@cp.loc <- c(0.25,0.75)
pw.acd.obj@lambda_0 <- c(1,2,1)
pw.acd.obj@alpha <- rep(0.2,3)
pw.acd.obj@beta <- rep(0.7,3)
pw.acd.obj@N <- 3000
pw.acd.obj <- pc_acdsim(pw.acd.obj)
ts.plot(pw.acd.obj@x)
ts.plot(pw.acd.obj@psi)
```

simHawkes-class *An S4 class for a nonstationary ACD model.*

Description

A specification class to create an object of a simulated piecewise constant Hawkes model of order (1,1). We consider the following time-varying piecewise constant Hawkes process (which we term tvHawkes) $\lambda(\upsilon) = \lambda_0(\upsilon) + \sum_{\upsilon_t < s} \alpha(\upsilon) e^{-\beta(\upsilon)(\upsilon - \upsilon_t)}$, for $\upsilon = 1, \ldots, T$.

Value

Returns an object of simHawkes class.

Slots

H The durational time series.

cH The psi time series.

horizon The time horizon of a Hawkes process typically expressed in seconds. Effective sample size will differ depending on the size of the parameters.

N Effective sample size which differs depending on the size of the parameters.

cp.loc The vector with the location of the changepoints. Takes values from 0 to 1 or NULL if none. Default is NULL.

lambda_0 The vector of the parameters lambda_0 in the Hawkes model as in the above formula.

alpha The vector of the parameters alpha in the Hawkes model as in the above formula.

beta The vector of the parameters beta in the Hawkes model as in the above formula.

References

Korkas, K.K., 2022. Ensemble binary segmentation for irregularly spaced data with change-points. Journal of the Korean Statistical Society, 51(1), pp.65-86.

Examples

```
pw.hawk.obj <- new("simHawkes")
pw.hawk.obj@cp.loc <- c(0.5)
pw.hawk.obj@lambda_0 <- c(1,2)
pw.hawk.obj@alpha <- c(0.2,0.2)
pw.hawk.obj@beta <- c(0.7,0.7)
pw.hawk.obj@horizon <- 1000
pw.hawk.obj <- pc_hawkessim(pw.hawk.obj)
ts.plot(pw.hawk.obj@H)
ts.plot(pw.hawk.obj@cH)
```

Z_trans *Transformation of an irregularly spaces time series.*

Description

Transformation of a irregularly spaces time series. For the tvACD model, we calculate $U_t = g_0(x_t, \psi_t) = \frac{x_t}{\psi_t}$, where $\psi_t = C_0 + \sum_{j=1}^{p} C_j x_{t-j} + \sum_{k=1}^{q} C_{p+k}\psi_{t-k} + \epsilon x_t$. where the last term ϵx_t is added to ensure the boundness of U_t.

Usage

```
Z_trans(
  H,
  start.values = c(0.9, 0.6),
  dampen.factor = "auto",
  epsilon = 1e-05,
  LOG = TRUE,
  process = "acd",
  acd_p = 0,
  acd_q = 1
)

## S4 method for signature 'ANY'
Z_trans(
  H,
  start.values = c(0.9, 0.6),
  dampen.factor = "auto",
  epsilon = 1e-05,
  LOG = TRUE,
  process = "acd",
  acd_p = 0,
  acd_q = 1
)
```

Arguments

H	The input irregular time series.
start.values	Warm starts for the optimizers of the likelihood functions.
dampen.factor	The dampen factor in the denominator of the residual process. Default is "auto".
epsilon	A parameter added to ensure the boundness of the residual process. Default is 1e-6.
LOG	Take the log of the residual process. Default is TRUE.
process	Choose between acd or hawkes. Default is acd.
acd_p	The p order of the ACD model. Default is 0.
acd_q	The q order of the ACD model. Default is 1.

Value

Returns the transformed residual series.

References

Korkas, K.K., 2022. Ensemble binary segmentation for irregularly spaced data with change-points. Journal of the Korean Statistical Society, 51(1), pp.65-86.

Examples

```
pw.acd.obj <- new("simACD")
pw.acd.obj@cp.loc <- c(0.25,0.75)
pw.acd.obj@lambda_0 <- c(1,2,1)
pw.acd.obj@alpha <- rep(0.2,3)
pw.acd.obj@beta <- rep(0.7,3)
pw.acd.obj@N <- 1000
pw.acd.obj <- pc_acdsim(pw.acd.obj)
ts.plot(Z_trans(pw.acd.obj@x))
```

Index