Package ‘L0cpt’

March 23, 2026

Type Package

Title Change Point Detection with L0 Penalty

Version 0.2.0

Date 2026-3-18

Description
Under an L0 penalty framework, a computationally efficient implementation of change point detection is developed. By integrating active set algorithms with warm start initialization, the package achieves linear-time complexity for solving change point detection problems. References: Wen et al. (2020) <doi:10.18637/jss.v094.i04>; Zhu et al. (2020)<doi:10.1073/pnas.2014241117>.

License GPL (>= 3)

RoxygenNote 7.3.2

Encoding UTF-8

Imports ggplot2, stats

Suggests knitr, rmarkdown

NeedsCompilation no

Author Tianhao Wang [aut, cre]

Maintainer Tianhao Wang <tianhaowang@mail.ustc.edu.cn>

Repository CRAN

Date/Publication 2026-03-23 17:20:28 UTC

Contents

`L0cpt-package` *A package for L0 constrained change point approximation*

Description

L0 penalty is useful for change point detection, commonly referred to as L0 penalized and L0 constraint problems. The L0 penalized problem which applies the hyperparameter λ, is illustrated below:

$$\underset{1=t_0<t_1<\cdots<t_K<t_{K+1}=n+1}{\arg\min} \sum_{k=1}^{K+1} \sum_{i=t_{k-1}}^{t_k-1} \left(y_i - \bar{y}_{(t_{k-1}:t_k-1)}\right)^2 + \lambda K$$

In practice, rather than selecting λ within an appropriate range, the number of change points (a positive integer $\boldsymbol{s}$) is more feasible and convenient in the L0 constraint model:

$$\min_{1=t_0<t_1<\cdots<t_{\hat{K}}<t_{\hat{K}+1}=n+1} \sum_{k=1}^{\hat{K}+1} \sum_{i=t_{k-1}}^{t_k-1} \left(y_i - \bar{y}_{(t_{k-1}:t_k-1)}\right)^2, \quad \text{subject to } \hat{K} = \boldsymbol{s}$$

For such L0 constraint problems, we employ a splicing-based approach to design algorithms for processing. This package has the following five main methods:

With fixed change points Fit a piecewise constant estimated trend with a given number of change points.

With optimal change points Fit a piecewise constant estimated trend with a maximum number of change points, and select the optimal estimated trend using appropriate information criteria.

Simulated data Generate piecewise constant data.

Print/coef Print a summary of the change point detection and trend estimation results.

Plot Plot a summary of the trend estimation results.

Details

_PACKAGE

- Due to the connection between L0 constraint problems and L0 penalty problems, and considering that the sparsity of change points is is more meaningful in practical applications than the selection of the hyperparameter λ. We focus on the constraint that reflects our aim to achieve the estimated trend and change-point estimator with a given number of change points.

References

Harchaoui Z and Lévy-Leduc C. Multiple change-point estimation with a total variation penalty. Journal of the American Statistical Association (2010).

Qian J and Su L. Shrinkage estimation of regression models with multiple structural. Econometric Theory (2016).

coef.L0cptfix *Extract the fitted coefficients of beta*

Description

Extracts the coefficients of the estimated beta under the constraint of a given number of change points.

Usage

```
## S3 method for class 'L0cptfix'
coef(object, ...)
```

Arguments

`object`	The output of L0cptfix
`...`	ignore

Value

The non-zero coefficient values of the vector beta in the process.

Examples

```
n = 500
sigma = 0.7
tau = c(0.1, 0.25, 0.3, 0.4, 0.7, 0.85, 0.95)
h = c(-1, 5, 3, 0, -1, 2, 0, -1)
seed = 50
blocksdata = SimuBlocks(n, sigma, seed, tau, h)

k = 7
first = 0
last = 1
blocksdatafit_fix = L0cpt.fix(blocksdata$y, k, first, last)
coef(blocksdatafit_fix)
```

coef.L0cptopt *Extract the optimal coefficients of beta*

Description

Extracts the coefficients of the estimated beta under the optimal change points.

Usage

```
## S3 method for class 'L0cptopt'
coef(object, ...)
```

Arguments

`object`	The output of L0cptopt
`...`	ignore

Value

The non-zero coefficient values of the vector beta in the process.

Examples

```
n = 500
sigma = 0.7
tau = c(0.1, 0.25, 0.3, 0.4, 0.7, 0.85, 0.95)
h = c(-1, 5, 3, 0, -1, 2, 0, -1)
seed = 50
blocksdata = SimuBlocks(n, sigma, seed, tau, h)

kmax = 15
first = 0
last = 1
blocksdatafit_opt = L0cpt.opt(blocksdata$y, kmax, first, last, "bic")
coef(blocksdatafit_opt)
```

`L0cpt.fix` *The L0 change-point detection method with fixed change points*

Description

Fit the input data points to a piecewise constant trend with a given number of change points.

Usage

```
L0cpt.fix(y = y, k = k, first = 0, last = 1, M = 5)
```

Arguments

`y`	The input data points
`k`	The given number of change points
`first`	The value ranges from 0 to 1. Represent the minimum percentile point where a change point may occur. If 'first' = 0.01, it means that change points cannot appear in the first 1% of the data points. If 'first' = 0, there is no constraint on the position of the change point.

`last`	The value ranges from 0 to 1. Represent the maximum percentile point where a change point may occur. If 'last' = 0.99, it means that change points cannot appear in the last 1% of the data points. If 'last' = 1, there is no constraint on the position of the change point.
`M`	The maximum number of exchange change-points in the splicing method. In the vast majority of cases, $M = 5$ is sufficient. If very high precision is desired, $M = \frac{k}{3}$ or $\frac{k}{4}$ can be considered.

Value

An S3 object of type "L0cptfix". A list containing the fitted trend results:

`y`	The input data points
`betak`	The fitted $\hat{\beta}$ coefficients with the number of change points being k
`yk`	The fitted trend with the number of change points being k
`Ak`	The set of position indicators of the fitted change points with the number of change points being k

See Also

`L0cpt.opt`

Examples

```
tau = c(0.1, 0.3, 0.4, 0.7, 0.85)
h = c(-1, 5, 3, 0, -1, 2)
BlocksData <- SimuBlocks(n = 350, sigma = 0.2, seed = 50, tau = tau ,h = h)
res <- L0cpt.fix(y=BlocksData$y, k=5, first=0.01, last=1, M=5)
print(res$Ak)
print(BlocksData$setA)
plot(BlocksData$x, BlocksData$y, xlab="", ylab="")
lines(BlocksData$x, BlocksData$y0, col = "red")
lines(BlocksData$x, res$yk, col = "lightgreen")
```

`L0cpt.opt` *The L0 change-point detection method with optimal change points*

Description

Fit the input data points to a piecewise constant trend with optimal change points.

Usage

```
L0cpt.opt(y = y, kmax = kmax, first = 0, last = 1, penalty = "bic", M = 5)
```

Arguments

y	The input data points
kmax	The maximum number of change points
first	The value ranges from 0 to 1. Represent the minimum percentile point where a change point may occur. If 'first' = 0.01, it means that change points cannot appear in the first 1% of the data points. If 'first' = 0, there is no constraint on the position of the change point.
last	The value ranges from 0 to 1. Represent the maximum percentile point where a change point may occur. If 'last' = 0.99, it means that change points cannot appear in the last 1% of the data points. If 'last' = 1, there is no constraint on the position of the change point.
penalty	'sic' or 'bic' penalty
M	The maximum number of exchange change-points in the splicing method. In the vast majority of cases, $M = 5$ is sufficient. If very high precision is desired, $M = \frac{kmax}{3}$ or $\frac{kmax}{4}$ can be considered.

Details

Let the fitted trend be denoted as $\hat{\boldsymbol{y}}$, then

$$\text{sic} = n \times \log(\frac{1}{n}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2) + 2\log(\log(n)) \times \log(n) \times \text{df}(\hat{\boldsymbol{y}})$$

and

$$\text{bic} = n \times \log(\frac{1}{n}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2) + 2 \times \log(n) \times \text{df}(\hat{\boldsymbol{y}}).$$

The term $\text{df}(\hat{\boldsymbol{y}})$ represents the degrees of freedom for the estimated trend, where $\text{df}(\hat{\boldsymbol{y}}) = k + 1$. Here, k refers to the number of change points in the estimated trend.

Value

An S3 object of type "L0cptopt". A list containing the fitted trend results:

y	The input data points
betaopt	The fitted $\hat{\boldsymbol{\beta}}$ coefficients with optimal change points
yopt	The fitted trend with optimal change points
Aopt	The set of position indicators of the fitted change points with optimal change points
kopt	Optimal number of change points
ic	'sic' or 'bic' penalty.
kmax	The maximum number of change points.

Examples

```
tau = c(0.1, 0.3, 0.4, 0.7, 0.85)
h = c(-1, 5, 3, 0, -1, 2)
BlocksData <- SimuBlocks(n = 500, sigma = 0.2, seed = 50, tau = tau ,h = h)
res <- L0cpt.opt(y=BlocksData$y, kmax=20, first=0.01, last=1, penalty="bic", M=5)
print(res$Aopt)
print(BlocksData$setA)
plot(BlocksData$x, BlocksData$y, xlab="", ylab="")
lines(BlocksData$x, BlocksData$y0, col = "red")
lines(BlocksData$x, res$yopt, col = "lightgreen")
```

`plot.L0cptfix` *Plot L0cptfix object*

Description

Plots the estimated trend of L0cptfix object.

Usage

```
## S3 method for class 'L0cptfix'
plot(x, ...)
```

Arguments

`x`	The output of L0cptfix
`...`	ignore

Details

The vertical blue dashed line in the plot's x-axis represents the detected change point location.

Value

Plots a "L0cptfix" object.

Examples

```
n = 500
sigma = 0.7
tau = c(0.1, 0.25, 0.3, 0.4, 0.7, 0.85, 0.95)
h = c(-1, 5, 3, 0, -1, 2, 0, -1)
seed = 50
blocksdata = SimuBlocks(n, sigma, seed, tau, h)

k = 7
first = 0
last = 1
```

```
blocksdatafit_fix = L0cpt.fix(blocksdata$y, k, first, last)
plot(blocksdatafit_fix)
```

`plot.L0cptopt` *Plot L0cptopt object*

Description

Plots the optimal trend of L0cptopt object.

Usage

```
## S3 method for class 'L0cptopt'
plot(x, ...)
```

Arguments

`x`	The output of L0cptopt
`...`	ignore

Details

The vertical blue dashed line in the plot's x-axis represents the detected change point location.

Value

Plots a "L0cptopt" object.

Examples

```
n = 500
sigma = 0.7
tau = c(0.1, 0.25, 0.3, 0.4, 0.7, 0.85, 0.95)
h = c(-1, 5, 3, 0, -1, 2, 0, -1)
seed = 50
blocksdata = SimuBlocks(n, sigma, seed, tau, h)

kmax = 15
first = 0
last = 1
blocksdatafit_opt = L0cpt.opt(blocksdata$y, kmax, first, last, "bic")
plot(blocksdatafit_opt)
```

print.L0cptfix *Print L0cptfix object*

Description

Outputs the detected change point detection results.

Usage

```
## S3 method for class 'L0cptfix'
print(x, ...)
```

Arguments

`x`	The output of L0cptfix
`...`	ignore

Details

For clarity of presentation, we retain only four decimal places for the fitted change point positions normalized to the interval [0, 1].

Value

Prints a summary of the fitted change points in "L0cptfix" object.

Examples

```
n = 500
sigma = 0.7
tau = c(0.1, 0.25, 0.3, 0.4, 0.7, 0.85, 0.95)
h = c(-1, 5, 3, 0, -1, 2, 0, -1)
seed = 50
blocksdata = SimuBlocks(n, sigma, seed, tau, h)

k = 7
first = 0
last = 1
blocksdatafit_fix = L0cpt.fix(blocksdata$y, k, first, last)
print(blocksdatafit_fix)
```

`print.L0cptopt` *Print L0cptopt object*

Description

Outputs the optimal change point detection results.

Usage

```
## S3 method for class 'L0cptopt'
print(x, ...)
```

Arguments

`x`	The output of L0cptopt
`...`	ignore

Details

For clarity of presentation, we retain only four decimal places for the fitted change point positions normalized to the interval [0, 1].

Value

Prints a summary of the fitted change points in "L0cptopt" object.

Examples

```
n = 500
sigma = 0.7
tau = c(0.1, 0.25, 0.3, 0.4, 0.7, 0.85, 0.95)
h = c(-1, 5, 3, 0, -1, 2, 0, -1)
seed = 50
blocksdata = SimuBlocks(n, sigma, seed, tau, h)

kmax = 15
first = 0
last = 1
blocksdatafit_opt = L0cpt.opt(blocksdata$y, kmax, first, last, "bic")
print(blocksdatafit_opt)
```

SimuBlocks — *Simulate Blocks Data*

Description

This function generates data points of piecewise constant trends.

Usage

```
SimuBlocks(n, sigma, seed = NA, tau, h)
```

Arguments

n	Number of data points
sigma	Standard deviation of the noise added to the signal
seed	An optional seed for random number generation to make results reproducible
tau	The locations of change points in the underlying trend
h	The constant values of the $length(tau) + 1$ segments of the underlying trend

Details

- To simplify the analysis, normalize the change point positions to a range between 0 and 1. Require that all elements of the input tau are within this range. Consequently, the change point positions in simulated data forms a subset of the set { $\frac{1}{n}$, $\frac{2}{n}$, $\frac{3}{n}$,. . . , 1}.
- In fact, $length(tau)$ change points can divide the interval into $length(tau) + 1$ segments of constant function values. Therefore, ensure that the length of vector h is $length(tau) + 1$.

Value

A list containing the piecewise constant simulated data and the underlying trend:

x	The set { $\frac{1}{n}$, $\frac{2}{n}$, $\frac{3}{n}$,. . . , 1}
y	The piecewise constant simulated data of length n
y0	The underlying trend of length n
setA	The set of position indicators of change points in the simulated data
tau	The locations of change points in the underlying trend

Examples

```
tau = c(0.1, 0.3, 0.4, 0.7, 0.85)
h = c(-1, 5, 3, 0, -1, 2)
BlocksData <- SimuBlocks(n = 350, sigma = 0.1, seed = 50, tau = tau ,h = h)
plot(BlocksData$x, BlocksData$y, xlab="", ylab="", col="grey", lty="blank", type="p")
lines(BlocksData$x, BlocksData$y0, col="black", lty="solid", type="l")
print(BlocksData$setA)
print(BlocksData$tau)
```

Index