Package 'L0TFinv'

May 7, 2026

Type Package

Title A Splicing Approach to the Inverse Problem of L0 Trend Filtering

Version 0.1.0

Date 2025-5-25

Description Trend filtering is a widely used nonparametric method for knot detection. This package provides an efficient solution for L0 trend filtering, avoiding the traditional methods of using Lagrange duality or Alternating Direction Method of Multipliers algorithms. It employ a splicing approach that minimizes L0-regularized sparse approximation by transforming the L0 trend filtering problem. The package excels in both efficiency and accuracy of trend estimation and changepoint detection in segmented functions. References: Wen et al. (2020) <doi:10.18637/jss.v094.i04>; Zhu et al. (2020)<doi:10.1073/pnas.2014241117>; Wen et al. (2023)

License GPL (>= 3)

RoxygenNote 7.3.2

Encoding UTF-8

Imports ggplot2, Matrix, stats

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

URL https://github.com/C2S2-HF/InverseL0TF

NeedsCompilation no

Author Tianhao Wang [aut, cre],
Canhong Wen [aut]

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

Repository CRAN

Date/Publication 2025-06-10 09:10:11 UTC

Contents

`L0TFinv-package` *A package for L0-regularized sparse approximation*

Description

Trend filtering is a typical method for nonparametric regression. The commonly used trend filtering models is the L1 trend filtering model (a) based on the difference matrix $\boldsymbol{D}^{(q+1)}$, as illustrated below.

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{D}^{(q+1)}\boldsymbol{\alpha}\|_{\ell_1}, \quad q = 0, 1, 2, \ldots. \quad (a)$$

L0 trend filtering (b) has a advantage over other trend filtering methods, especially in the detection of change points. The expression for L0 trend filtering is as follows:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{D}^{(q+1)}\boldsymbol{\alpha}\|_{\ell_0}. \quad (b)$$

We explore transforming the problem (b) into a L0-regularized sparse format (c) by introducing an artificial design matrix $\boldsymbol{X}^{(q+1)}$ that corresponds to the difference matrix, thereby reformulating the L0 trend filtering problem into the following format.

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^n} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}^{(q+1)}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{i=q+2}^{n} |\boldsymbol{\beta}_i|_{\ell_0}. \quad (c)$$

In our practical approach, we consider the maximum number of change points $k_{\max}$ as a constraint, transforming the aforementioned L0 penalty problem (c) into the following L0 constraint problem.

$$\text{minimize } \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}^{(q+1)}\boldsymbol{\beta}\|_2^2, \quad \text{subject to } \sum_{i=q+2}^{n} |\boldsymbol{\beta}_i|_{\ell_0} \le k_{\max}. \quad (d)$$

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

matrix with special structure Generate $\boldsymbol{X}^{(q+1)}$ or $\boldsymbol{D}^{(q+1)}$ matrix.

inverse of the crossprod matrix Simplify the calculation of the inverse matrix of $(\boldsymbol{X}_A^{(q+1)})^T\boldsymbol{X}_A^{(q+1)}$ for the cases where $q = 0$ or $q = 1$, which is frequently used in splicing algorithms.

inverse L0 trend filtering with fixed change points Fit a piecewise constant or piecewise linear estimated trend with a given number of change points.

inverse L0 trend filtering with optimal change points Fit a piecewise constant or piecewise linear estimated trend with a maximum number of change points, and select the optimal estimated trend using appropriate information criteria.

simulated data Generate piecewise constant or piecewise linear data.

print/coef Print a summary of the trend estimation results.

plot Plot a summary of the trend estimation results.

Details

_PACKAGE

- In previous studies, algorithms solving trend filtering problems (a) necessitate the computation of $((\boldsymbol{D}^{(q+1)})^T\boldsymbol{D}^{(q+1)})^{-1}$. When n is large, just fitting the matrix into memory becomes an issue.
- In L0 trend filtering (b), the positions of non-zero elements in the L0 norm correspond with the locations of change points. We consider two subsets: the active set A for non-zero elements and the inactive set I for zero elements. Despite this, computing $((\boldsymbol{D}_I^{(q+1)})^T\boldsymbol{D}_I^{(q+1)})^{-1}$ remains a task involving a substantial matrix.
- Due to the connection between L0 constraint problems and L0 penalty problems, and considering that the sparsity of $\boldsymbol{\beta}$ 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 an estimated trend with a given number of change points. So we transform the L0 penalty problem (c) into the L0 constraint problem (d).

References

Kim SJ, Koh K, Boyd SP and Gorinevsky DM. L1 Trend Filtering. Society for Industrial and Applied Mathematics (2009).

Wen C, Wang X and Zhang A. L0 Trend Filtering. INFORMS Journal on Computing (2023).

`coef.L0TFinvfix` *Extract estimated trends*

Description

Extract the coefficients of the estimated trends under the constraint of a given number of change points.

Usage

```
## S3 method for class 'L0TFinvfix'
coef(object, k = NULL, ...)

## S3 method for class 'L0TFinvopt'
coef(object, k = NULL, ...)
```

Arguments

object	The output of L0TFinvfix or L0TFinvopt
k	The given number of change points
...	ignore

Value

Estimated parameters and trends for L0TFinvfix or L0TFinvopt models with specified numbers of change points

See Also

print.L0TFinvfix

DiffMat *Generate a difference matrix*

Description

This function generates a matrix for computing differences of a certain order, useful in numerical methods and for creating specific matrix patterns.

Usage

```
DiffMat(n, q)
```

Arguments

n	The number of data points
q	The order of the difference

Value

A matrix with dimensions $n - q - 1$ by n, whose elements correspond to the combinatorial values of q.

See Also

XMat

Examples

```
Mat1 <- DiffMat(n = 10, q = 0)
print(Mat1)

Mat2 <- DiffMat(n = 15, q = 1)
print(Mat2)

Mat3 <- DiffMat(n = 15, q = 2)
print(Mat3)
```

`L0TFinv.fix` *The inverse L0 trend filtering with fixed change points*

Description

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

Usage

```
L0TFinv.fix(y = y, k = k, q = q, first = 0, last = 1)
```

Arguments

`y`	The input data points
`k`	The given number of change points
`q`	0 or 1. Correspond to a piecewise constant or piecewise linear trend
`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.

Value

An S3 object of type "L0TFinvfix". A list containing the fitted trend results:

`sic`	Information criterion value with a penalty term of $2\log(\log(n)) \times \log(n)$
`bic`	Information criterion value with a penalty term of $2 \times \log(n)$
`mse`	The mean square error between the fitted trend and the input data
`y`	The input data points

betak	The fitted $\hat{\boldsymbol{\beta}}$ 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
beta.all	A data frame with dimensions $n \times k$, where each column represents the fitted $\hat{\boldsymbol{\beta}}$ coefficients corresponding to a given number of change points
y.all	A data frame with dimensions $n \times k$, where each column represents the fitted estimated trend corresponding to a given number of change points
A.all	A list of length k, where each element corresponds to the set of position indicators of change points under a given number

See Also

L0TFinv.opt

Examples

```
tau = c(0.1, 0.3, 0.4, 0.7, 0.85)
h = c(-1, 5, 3, 0, -1, 2)
BlocksData <- SimuBlocksInv(n = 350, sigma = 0.2, seed = 50, tau = tau ,h = h)
res <- L0TFinv.fix(y=BlocksData$y, k=5, q=0, first=0.01, last=1)
print(res$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")

tau1 = c(0.1, 0.3, 0.4, 0.7, 0.85)
h1 = c(-1, 5, 3, 0, -1, 2)
a0 = -10
WaveData <- SimuWaveInv(n = 2000, sigma = 0.1, seed = 50, tau = tau1, h = h1, a0 = a0)
res1 <- L0TFinv.fix(y=WaveData$y, k=5, q=1, first=0, last=0.99)
print(res1$Ak)
print(WaveData$setA)
plot(WaveData$x, WaveData$y, xlab="", ylab="")
lines(WaveData$x, WaveData$y0, col = "red")
lines(WaveData$x, res1$yk, col = "lightgreen")
```

L0TFinv.opt *The inverse L0 trend filtering with optimal change points*

Description

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

Usage

```
L0TFinv.opt(y = y, kmax = kmax, q = q, first = 0, last = 1, penalty = "bic")
```

Arguments

y	The input data points
kmax	The maximum number of change points
q	0 or 1. Correspond to a piecewise constant or piecewise linear trend
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

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 $\text{df}(\hat{\boldsymbol{y}}) = k + q + 1$. Here, k refers to the number of change points in the estimated trend.

Value

An S3 object of type "L0TFinvopt". A list containing the fitted trend results:

sic	Information criterion value with a penalty term of $2\log(\log(n)) \times \log(n)$
bic	Information criterion value with a penalty term of $2 \times \log(n)$
mse	The mean square error between the fitted trend and the input data
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
beta.all	A data frame with dimensions $n \times k_{\max}$, where each column represents the fitted $\hat{\boldsymbol{\beta}}$ coefficients corresponding to a given number of change points

y.all	A data frame with dimensions $n \times k_{\max}$, where each column represents the fitted estimated trend corresponding to a given number of change points
A.all	A list of length $k_{\max}$, where each element corresponds to the set of position indicators of change points under a given number

Examples

```
tau = c(0.1, 0.3, 0.4, 0.7, 0.85)
h = c(-1, 5, 3, 0, -1, 2)
BlocksData <- SimuBlocksInv(n = 500, sigma = 0.2, seed = 50, tau = tau ,h = h)
res <- L0TFinv.opt(y=BlocksData$y, kmax=20, q=0, first=0.01, last=1, penalty="bic")
print(res$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")

tau1 = c(0.4, 0.6, 0.7)
h1 = c(-3, 5, -4, 6)
a0 = -10
WaveData <- SimuWaveInv(n = 500, sigma = 0.1, seed = 50, tau = tau1, h = h1, a0 = a0)
res1 <- L0TFinv.opt(y=WaveData$y, kmax=10, q=1, first=0, last=0.99, penalty="sic")
print(res1$Aopt)
print(WaveData$setA)
plot(WaveData$x, WaveData$y, xlab="", ylab="")
lines(WaveData$x, WaveData$y0, col = "red")
lines(WaveData$x, res1$yopt, col = "lightgreen")
```

---

plot.L0TFinvfix *Plot L0TFinvfix or L0TFinvopt object*

Description

Plots a summary of L0TFinvfix or L0TFinvopt

Usage

```
## S3 method for class 'L0TFinvfix'
plot(x, type = NULL, k = NULL, ...)

## S3 method for class 'L0TFinvopt'
plot(x, type = "mse", k = NULL, ...)
```

Arguments

x	The output of L0TFinvfix or L0TFinvopt

type	The values are taken as c("mse", "sic", "bic", "$yhat$"). If $type$ is "mse", plot the mse as it changes with change points. The same applies to "sic" and "bic". If $type$ is "$yhat$", plot the trend of the estimated values against the input data.
k	Only used for $type$ = "$yhat$". The given number of change points. By default, the L0TFinvfix object outputs the estimated trend that corresponds to the fixed number of change points within the model. Conversely, the L0TFinvopt object provides the estimated trend based on the optimal change points.
...	ignore

Value

Plot the information criteria for L0TFinvfix and L0TFinvopt across an increasing number of change points or display the estimated trend with a selected change point

See Also

print.L0TFinvfix

---

print.L0TFinvfix *Print L0TFinvfix or L0TFinvopt object*

Description

Prints a summary of L0TFinvfix or L0TFinvopt

Usage

```
## S3 method for class 'L0TFinvfix'
print(x, ...)

## S3 method for class 'L0TFinvopt'
print(x, ...)
```

Arguments

x	The output of L0TFinvfix or L0TFinvopt
...	ignore

Value

Estimated parameters and information criteria for L0TFinvfix and L0TFinvopt across an increasing number of change points

Examples

```
library(ggplot2)

tau = c(0.1, 0.3, 0.4, 0.7, 0.85)
h = c(-1, 5, 3, 0, -1, 2)
BlocksData <- SimuBlocksInv(n = 500, sigma = 0.2, seed = 50, tau = tau ,h = h)
res <- L0TFinv.opt(y=BlocksData$y, kmax=10, q=0, first=0.01, last=1, penalty="bic")
print(res)
coef(res,k=res$kopt)
plot(res,type="yhat")
plot(res,type="bic")

tau1 = c(0.1, 0.3, 0.4, 0.7, 0.85)
h1 = c(-1, 5, 3, 0, -1, 2)
a0 = -10
WaveData <- SimuWaveInv(n = 2000, sigma = 0.1, seed = 50, tau = tau1, h = h1, a0 = a0)
res1 <- L0TFinv.fix(y=WaveData$y, k=20, q=1, first=0, last=0.99)
print(res1)
coef(res1,k=5)
plot(res1,type="yhat",k=5)
plot(res1,type="mse")
```

`SimuBlocksInv` *Simulate Blocks Data*

Description

This function generates data points of piecewise constant trends.

Usage

```
SimuBlocksInv(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 $\{ \frac{1}{n}, \frac{2}{n}, \frac{3}{n}, \ldots, 1\}$.
- 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 $\{ \frac{1}{n}, \frac{2}{n}, \frac{3}{n}, \ldots, 1\}$
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 <- SimuBlocksInv(n = 350, sigma = 0.1, seed = 50, tau = tau ,h = h)
plot(BlocksData$x, BlocksData$y, xlab="", ylab="")
lines(BlocksData$x, BlocksData$y0, col = "red")
print(BlocksData$setA)
print(BlocksData$tau)
```

SimuWaveInv *Simulate Wave Data*

Description

This function generates data points of piecewise linear trends.

Usage

```
SimuWaveInv(n, sigma, seed = NA, tau, h, a0 = 0)
```

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 slope of the $length(tau) + 1$ segments of the underlying trend
a0	The initial point value

Value

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

x	The set $\{ \frac{1}{n}, \frac{2}{n}, \frac{3}{n}, \ldots, 1 \}$
y	The piecewise linear 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

See Also

SimuBlocksInv

Examples

```
tau = c(0.1, 0.3, 0.4, 0.7, 0.85)
h = c(-1, 5, 3, 0, -1, 2)
a0 = -10
WaveData <- SimuWaveInv(n = 650, sigma = 0.1, seed = 50, tau = tau, h = h, a0 = a0)
plot(WaveData$x, WaveData$y, xlab="", ylab="")
lines(WaveData$x, WaveData$y0, col = "red")
print(WaveData$setA)
print(WaveData$tau)
```

---

solMat *Generate the inverse of the crossprod matrix*

Description

Generate the inverse matrix of $(\boldsymbol{X}_A^{(q+1)})^T \boldsymbol{X}_A^{(q+1)}$ for the cases where $q = 0$ or $q = 1$, commonly employed in splicing algorithms. Note that an explicit solution exists for the inverse when $q = 0$, but not when $q = 1$.

Usage

```
solMat(n, q, A)
```

Arguments

n	The number of data points
q	The order of the difference, 0 or 1
A	The set of indicators, a subset of $\{1, 2, 3, \ldots, n\}$

Value

The inverse matrix of $(\boldsymbol{X}_A^{(q+1)})^T \boldsymbol{X}_A^{(q+1)}$ for the cases where $q = 0$ or 1.

Examples

```
Mat1 <- XMat(n = 10, q = 0)
A1 = c(1,2,5,8)
mat1 = as.matrix(Mat1[,A1])
S1 <- solMat(n = 10, q = 0, A = A1)
print(S1)
print(round(S1%*%t(mat1)%*%mat1,10))

Mat2 <- XMat(n = 15, q = 1)
A2 = c(1,3,8,10,15)
```

```
mat2 = as.matrix(Mat2[,A2])
S2 <- solMat(n = 15, q = 1, A = A2)
print(S2)
print(round(S2%*%t(mat2)%*%mat2,10))
```

`TFmetrics` *Print four metrics about change point detection results*

Description

Prints four metrics to compare the quality of change point detection results.

Usage

```
TFmetrics(y0, tau = NULL, yhat, cpts = NULL)
```

Arguments

`y0`	The underlying trend
`tau`	The locations of change points in the underlying trend
`yhat`	The fitted trend
`cpts`	The positions of the fitted change points

Details

$\hat{\boldsymbol{\tau}}$ represents the estimated change point positions, while $\boldsymbol{\tau}$ denotes the locations of change points in the underlying trend.

$$d_H = \frac{1}{n} \max\{\max_k \min_j |\tau_j - \hat{\tau}_k|, \max_j \min_k |\tau_j - \hat{\tau}_k|\}.$$

Note that the number of $\hat{\boldsymbol{\tau}}$ and $\boldsymbol{\tau}$ does not need to be the same.

Value

`MSE`	The mean square error between the fitted trend and the underlying trend
`MAD`	The median absolute deviation between the fitted trend and the underlying trend
`dH`	Hausdorff Distance (dH) measures the accuracy of the estimated change points
`nknot`	The number of detected change points

Examples

```
tau = c(0.1, 0.3, 0.4, 0.7, 0.85)
h = c(-1, 5, 3, 0, -1, 2)
n = 500
BlocksData <- SimuBlocksInv(n = n, sigma = 0.2, seed = 50, tau = tau ,h = h)
res <- L0TFinv.opt(y=BlocksData$y, kmax=10, q=0, first=0.01, last=1, penalty="bic")
metrics <- TFmetrics(BlocksData$y0,BlocksData$tau,res$yopt,res$Aopt/n)
print(metrics)

tau1 = c(0.1, 0.3, 0.4, 0.7, 0.85)
h1 = c(-1, 5, 3, 0, -1, 2)
a0 = -10
n1 = 2000
WaveData <- SimuWaveInv(n = n1, sigma = 0.1, seed = 50, tau = tau1, h = h1, a0 = a0)
res1 <- L0TFinv.fix(y=WaveData$y, k=20, q=1, first=0, last=0.99)
metrics1 <- TFmetrics(WaveData$y0,WaveData$tau,res1$y.all[,5],res1$A.all[[5]]/n1)
print(metrics1)
```

`XMat` *Generate an artificial design matrix*

Description

This matrix corresponds to the difference matrix, transforming the L0 trend filtering model into an inverse statistical problem.

Usage

```
XMat(n, q)
```

Arguments

`n`	The number of data points
`q`	The order of the difference

Details

Noticing the correspondence between $\boldsymbol{D}^{(q+1)}$ and $\boldsymbol{X}^{(q+1)}$, the result of their matrix multiplication is a combination of a zero matrix and an identity matrix. Expressed as $\boldsymbol{D}^{(q+1)}\boldsymbol{X}^{(q+1)} = (\boldsymbol{O}_{(n-q-1)\times(q+1)}, \quad \boldsymbol{I}_{(n-q-1)\times(n-q-1)})$. The result is advantageous for the invertible processing of the original L0 trend filtering problem.

Value

A matrix with dimensions n by n, whose elements correspond to the difference matrix.

Examples

```
mat1 <- XMat(n = 10, q = 0)
print(mat1)

mat2 <- XMat(n = 15, q = 1)
print(mat2)

mat3 <- XMat(n = 15, q = 2)
print(mat3)

Mat1 <- DiffMat(n = 10, q = 0)
Mat2 <- DiffMat(n = 15, q = 1)
Mat3 <- DiffMat(n = 15, q = 2)
print(Mat1%*%mat1)
print(Mat2%*%mat2)
print(Mat3%*%mat3)
```

Index