Package 'AMIAS'

March 14, 2019

Type Package

Title Alternating Minimization Induced Active Set Algorithms

Version 1.1.0

Date 2019-3-13

Author Canhong Wen, Xueqin Wang, Shijie Quan, Zelin Hong, Aijun Zhang

Maintainer Canhong Wen <wench@ustc.edu.cn>

Depends utils,zoo,limSolve,Matrix

Description
An implementation of alternating minimization induced active set (AMIAS) method for solving the generalized L0 problem. The AMIAS method is based on the necessary optimality conditions derived from an augmented Lagrangian framework. The proposed method takes full advantage of the primal and dual variables with complementary supports, and decouples the high-dimensional problem into two sub-systems on the active and inactive sets, respectively. A sequential AMIAS algorithm with warm start initialization is developed for efficient determination of the cardinality parameter, along with the output of solution paths.

License LGPL-3

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-03-14 14:23:31 UTC

R topics documented:

amias *Solve Generalized l_0 Problem with AMIAS Method*

Description

The function solves the Generalized l_0 problem with a user-specified number of knots. The generalized l_0 coefficient is computed via the alternating minimization induced active set (AMIAS) algorithm. Can deal with any polynomial order or even a general penalty matrix for structural filtering.

Usage

```
amias(y, D = NULL, D_type = c("tf0", "tfq", "user"), q = 0, k = 3, rho = n^(q+1),
    tmax = 10, A = NULL, smooth = FALSE, h = 5, adjust = FALSE, delta = 10,
    adjust.max = 10,...)
```

Arguments

y	Observed data, of length n.
D	Penalty matrix of size m x n (see Details). Default is `NULL`.
D_type	Types of D. Either "tf0", "tfq" or "user", depending on what types of contraint that user wants to impose on the data. For `D_type = "tfq"` and `D = NULL`, we solve the l_0 trend filtering of order $q+1$, where q is determined by the argument `q`. For `D_type = "tf0"` and `D = NULL`, we solve the l_0 trend filtering of order 0, i.e., with piecewise constant constraint on the data. For `D_type = "user"`, the penalty matrix `D` must be specified.
q	Nonnegative integer used to specify the order of penalty matrix. See `genDtf1d` for Details.
k	The number of knots. It is also the size of active set `A` or the number of equality satisfied in the contraint $D\beta = 0$. Default is 3.
rho	The hyperparameter ρ in the augmented Lagrangian of the l_0 trend filtering problem. Default is n^{q+1}.
tmax	The maximum number of iterations in the AMIAS algorithm.
smooth	Whether to smooth the data, if `TRUE`, it smoothes the input data. Default is FALSE.
h	Bandwidth in smoothing data. See `my.rollmean` for details.
A	Initialization for the active set. Default is NULL, which corresponds to an empty set.
adjust	Whether to adjust the indexes of the active set in the AMIAS algorithm. If `TRUE`, it implements the adjustment when the indexes in the active set are not well separated. Default is FALSE.
delta	The minimum gap etween the adjacent knots. Only used when `adjust = TRUE`.
adjust.max	The number of iterations in the adjustment step. Only used when `adjust = TRUE`.
...	Other arguments.

Details

The generalized l_0 problem with a user-specified number of knots k is

$$\min_\alpha 1/2\|y-\alpha\|_2^2 \text{ s.t. } \|D\alpha\|_0 = k.$$

The penalty matrix D is either the discrete difference operator of order $q+1$ or a general matrix specified by users to enforce some special structure.

The generalized l_0 problem is solved via the alternating minimization induced active set (AMIAS) method, which is proposed by Wen et al. (2018). In AMIAS, by the augmented Lagrangian formulation with a variable splitting technique, we consider the associated augmented Lagrangian:

$$\min_\alpha 1/2\|y-\alpha\|_2^2 + u^T(D\alpha - v) + \rho/2\|D\alpha - v\|_2^2; \text{ s.t. } \|v\|_0 = k.$$

Based on this formulation, we derive the necessary optimality conditions through alternating minimization and coordinatewise hard thresholding. The conditions are based on primal variable v and dual variable u with complementary supports, namely active and inactive sets (denoted by A and I). Given A, I, the KKT conditions on the high dimensional variables are decoupled into two sub-systems with elegant linear algebraic solutions. For more details, please see Wen et al. (2019).

When D is the discrete difference operator of order $q+1$, it reduces to the l_0 trend filtering problem, which will produce a piecewise q-th polynomial curve with automatically identified knots. In this case, the penalty matrix is a banded matrix and thus banded Cholesky decomposition is applied to accelerate the computation. In particular, when $q = 0$, it produces a piecewise constant estimate with automatically detected change points. We use a fast and memory-saving strategy to further improve the computational efficiency.

To prevent the adjacent detected knots not to be too close, an adjustment step can be added in each step of the AMIAS method. When an adjustment is used, users need to specify the minimum possible gap between the adjacent knots and the maximum number of iterations in one adjustment step.

Value

A list with class attribute 'amias' and named components:

`call`	The call that produces this object.
`y`	Observed sequence, if smooth, the smooth y will be returned.
`D_type`	Types of D.
`q`	The order of penalty matrix.
`k`	The number of knots.
`alpha`	The fitting coefficients α.
`v`	The primal variable or splitting variable of the argumented lagrangian form in $D\alpha$.
`u`	The dual variable or lagrangian operator of the argumented lagrangian form in $D\alpha$ for linear item.
`A`	The final estimate of active set, i.e., set of the detected knots.
`df`	Degree of freedom of the fitting model, which is defined as $df = k + q + 1$.
`iter`	The iterations used.
`smooth`	Whether to smooth the data.

Author(s)

Canhong Wen, Xueqin Wang, Shijie Quan, Zelin Hong and Aijun Zhang.

Maintainer: Canhong Wen <wench@ustc.edu.cn>

References

Wen, C., Zhu, J., Wang, X., and Zhang, A. (2019) *L0 trend filtering*, technique report.

See Also

`print.amias`, `coef.amias` and `plot.amias` method, and the `samias` function.

Examples

```
##----- A toy example of piecewise constant signal -------
set.seed(0)
n <- 100
x = seq(1/n, 1,length.out = n)
y0 = 0*x; y0[x>0.5] = 1
y = y0 + rnorm(n, sd = 0.1)
fit <- amias(y, k = 1)
plot(fit)


##----- A toy example of piecewise linear trend -------
set.seed(0)
y0 = 2*(0.5-x); y0[x>0.5] = 2*(x[x>0.5]-0.5)
y = y0 + rnorm(n, sd = 0.1)
fit <- amias(y, D_type = "tfq", q = 1, k = 1)
print(fit)


##------ Piecewise constant trend filtering example in Wen et al.(2018).-----
set.seed(1)
data <- SimuBlocks(2048, sigma=0.5)

# with user-specified initialized active set (Here we use the true active set for initialization)
fit <- amias(data$y, k = 11, A = data$SetA)
plot(fit)
lines(data$x, data$y0, type="s") # Add the true signal for reference

# with adjustment
fit <- amias(data$y, k = 11, adjust = TRUE, delta = 20)
plot(fit, main = "Blocks")
lines(data$x, data$y0, type="s") # Add the true signal for reference


##------ Piecewise linear trend filtering example in Wen et al.(2018).-----
set.seed(1)
data <- SimuWave(512)
fit <- amias(data$y, q = 1, D_type = "tfq", k = 8, adjust = TRUE, delta = 30)
```

```
plot(fit, main = "Wave")
lines(data$x, data$y0, type="l")


##------ Doppler example in Wen et al.(2018).-----
set.seed(1)
data <- SimuDoppler(1024)
A.ini <- sample(1024,30)
op <- par(mfrow=c(1,2))
fit1 <- amias(data$y, q = 2, D_type = "tfq", k = 30, A = A.ini) # piecewise quadratic
plot(fit1, main = "Doppler")
fit2 <- amias(data$y, q = 3, D_type = "tfq", k = 30, A = A.ini) # piecewise Cubic polynomial
plot(fit2, main = "Doppler")
par(op)
```

`coef.amias` *Extract cefficients from a amias or samias object*

Description

This function extracts coefficients from a amias or samias object.

Usage

```
## S3 method for class 'amias'
coef(object, type=c("coef","primal", "dual", "active"), ...)
## S3 method for class 'samias'
coef(object, type = c("coef","primal", "dual", "active"), k, ...)
```

Arguments

`object`	A list with class attribute 'amias' or 'samias'.
`type`	A character string, one of `"coef"`, `"primal"`, `"dual"`, or `"active"`, indicating whether generalized l_0 coefficients, primal coefficients, dual coefficients, or active sets, should be returned. Default is `"coef"`, which corresponds to the solution of the original generalized l_0 problem.
`k`	An integer vector of numbers of knots at which coefficients should be calculated. The user can choose to a subset of index from $1, 2, \ldots, kmax$; none are specified, then it returns the optimal coefficient determined by miniming the BIC value.
`...`	additional arguments passed to coef.

Value

Returns a list with the following components:

`alpha`	If the type is `"coef"`, a matrix containing the generalized l_0 coefficients, each column corresponding to a value of `k`.

v	If the type is "primal", a matrix containing the primal variables, each column corresponding to a value of k.
u	If the type is "dual", a matrix containing the dual variables, each column corresponding to a value of k.
A	If the type is "active", a list containing the active sets, each sublist corresponding to a value of k.
k	An integer that specifies the optimal k or an integer vector containing the sequence of k values corresponding to the columns of beta, v, and u.

See Also

amias, samias.

Examples

```
##----- A toy example -------
set.seed(0)
n <- 100
x = seq(1/n, 1,length.out = n)
y0 = 0*x; y0[x>0.5] = 1
y = y0 + rnorm(n, sd = 0.1)


# For 'amias' object
fit <- amias(y, k = 1)
coef(fit)                # extract the fitting coefficients
coef(fit, type="active") # extract the detected knots

# For 'samias' object
fit <- samias(y, kmax = 6)
coef(fit)                         # get the coefficients with the minimum BIC value
coef(fit, k = 3:5)             # get the coefficients that number of knots being 3, 4, and 5

coef(fit, type="active")          # get the active set with minimum BIC value
coef(fit, type="active", k = 3:5) # get the active sets that number of knots being 3, 4, and 5
```

---

genDXX *Generate D Matrix*

Description

Generate D matrix which is used to do polynomial trend filtering of order k

Usage

```
genDtf1d(k,n=NULL,full=FALSE)
genDtf2d(k,dim1=NULL,dim2=NULL,full=FALSE)
genD2d(dim1=NULL,dim2=NULL,full=FALSE)
```

Arguments

k	The order of polynomial trend filtering.
n	Length of observe sequence y for genDtf1d, only be used in the `full` mode.
dim1	The number of rows for genDtf2d, only be used in the `full` mode.
dim2	The number of columns for genDtf2d, only be used in the `full` mode.
full	Whether to return the full matrix, if TRUE, it gennerate a full matrix.

Details

In 1d case, D is used to do polynomial trend filtering of order k, and generated by

$$D_{tf,k} = D1d * D_{tf,k-1} fork >= 2,$$

where the nozero elements in each row is `(1,-2,1)` in D_tf,1 and `(-1,1)` in D_1d. This is usually not necessary to call directly, as `AMIAS` internally gennerate D, but inspection of the matrix can sometimes be useful.

Value

vecctor	Vector: return when full is FALSE, nozero elements in first row(equal to any other row) of D.
matrix	Matrix: return when full is TRUE, whole sparse D.

Author(s)

Canhong Wen, Xueqin Wang, Shijie Quan, Zelin Hong and Aijun Zhang.

Maintainer: Canhong Wen <wench@ustc.edu.cn>

References

Wen, C., Zhu, J., Wang, X., and Zhang, A. (2019) *L0 trend filtering*, technique report.

Examples

```
genDtf1d(k = 2)
genDtf1d(k = 1, n = 10, full = TRUE)
```

my.rollmean *Rolling Means*

Description

Generic functions for computing rolling means of ordered observations.

Usage

```
my.rollmean(y, h = 5 ,...)
```

Arguments

y	Observe sequence, with length n.
h	integer width of the rolling window, `rollmean` would be used with `k = h*2+1`.
...	Other arguments, see `rollmean` for details.

Value

vector	Vector: smooth observe sequence, with length n.

Author(s)

Canhong Wen, Xueqin Wang, Shijie Quan, Zelin Hong and Aijun Zhang.

Maintainer: Canhong Wen <wench@ustc.edu.cn>

References

Wen, C., Zhu, J., Wang, X., and Zhang, A. (2019) *L0 trend filtering*, technique report.

plot.amias *Plotting method for amias and samias objects*

Description

The function `plot.amias` plots the coefficient from an `"amias"` object with a user-specified number of knots as a function of the input positions (which are assumed to be evenly spaced if not specified). The function `plot.samias` produces either a profile plot of the solution path for the primal variable or the optimal fitted coefficients (which, recall, are determined by minimizing the Bayesian Information Criterion) from a fitted `"samias"` object.

Usage

```
## S3 method for class 'amias'
plot(x, add.knots = TRUE,...)
## S3 method for class 'samias'
plot(x, type = c("coef", "vpath"), k, add.label = TRUE, add.knots = TRUE,...)
```

Arguments

`x`	A list with class attribute 'amias' or 'samias'.
`add.knots`	Whether to add the locations of the detected knots at the top of the figure. It only works when `type = "coef"` for the function `plot.samias`.
`type`	Types of the plotting figure, either "coef" or "vpath". For `type = "coef"` and missing `k`, we plot the optimal l_0 fitted coefficients as well as the raw data. For `type = "coef"` and a user-specified `k`, we plot the l_0 fitted coefficients with number of knots `k` as well as the raw data. For `type = "vpath"`, we plot the solution path of the primal variable v versus the cardinality k, i.e., number of the detected knots.
`k`	The number of knots. Users can choose `k` from $1, 2, \ldots, kmax$; if none are specified, then it plots the optimal coefficient determined by miniming the BIC value.
`add.label`	Whether to add labels at the right side to indicate the locations of the detected knots. It only works when `type = "vpath"` for the function `plot.samias`.
`...`	Other arguments,see `plot.default` for details.

Author(s)

Canhong Wen, Xueqin Wang, Shijie Quan, Zelin Hong and Aijun Zhang.

Maintainer: Canhong Wen <wench@ustc.edu.cn>

References

Wen, C., Zhu, J., Wang, X., and Zhang, A. (2019) *L0 trend filtering*, technique report.

See Also

`amias`, `samias`.

Examples

```
##----- A toy example of piecewise constant signal -------
set.seed(0)
n <- 100
x = seq(1/n, 1,length.out = n)
y0 = 0*x; y0[x>0.5] = 1
y = y0 + rnorm(n, sd = 0.1)
```

```
# For 'amias' object
fit <- amias(y, k = 1)
plot(fit)

# For 'samias' object
fit <- samias(y, kmax = 5)
plot(fit, type = "coef", main = "Piecewise Constant")

op <- par(mfrow=c(1,2))
plot(fit, type= "coef", add.knots = FALSE, main = "Piecewise Constant")
plot(fit, type = "vpath", main = "Piecewise Constant")
par(op)

##----- A toy example of piecewise linear trend -------
set.seed(0)
y0 = 2*(0.5-x); y0[x>0.5] = 2*(x[x>0.5]-0.5)
y = y0 + rnorm(n, sd = 0.1)

# For 'amias' object
fit <- amias(y, D_type = "tfq", q = 1, k = 1)
plot(fit, main = "Piecewise Linear")

# For 'samias' object
fit <- samias(y, D_type = "tfq", q = 1, kmax = 4)
op <- par(mfrow=c(1,2))
plot(fit, type = "coef", main = "Piecewise Linear")
plot(fit, type = "vpath", main = "Piecewise Linear")
par(op)
```

`print.amias` *Print an amias or a samias object*

Description

Print a summary of the fitted "amias" or "samias" object.

Usage

```
## S3 method for class 'amias'
print(x, ...)
## S3 method for class 'samias'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	A list with class attribute 'amias' or 'samias'.
digits	significant digits in printout
...	Additional print arguments.

Details

For an "amias" object, the call that produced the object x is printed, followed by a summary of the output.

For a "samias" object, the call that produced the object x is printed, followed by a three-column matrix with columns k, Df and %BIC. The k column is the number of knots and the Df column is the degree of freedom defined as $Df = k + q + 1$. BIC is the BIC value.

Value

A summary or a three-column matrix is silently returned.

Author(s)

Canhong Wen, Xueqin Wang, Shijie Quan, Zelin Hong and Aijun Zhang.

Maintainer: Canhong Wen <wench@ustc.edu.cn>

References

Wen, C., Zhu, J., Wang, X., and Zhang, A. (2018) *L0 trend filtering*, technique report.

See Also

amias, samias.

Examples

```
##----- A toy example -------
set.seed(0)
n <- 100
x = seq(1/n, 1,length.out = n)
y0 = 0*x; y0[x>0.5] = 1
y = y0 + rnorm(n, sd = 0.1)

# For 'amias' object
fit <- amias(y, k = 1)
print(fit)

# For 'samias' object
fit <- samias(y, kmax = 5)
print(fit)
```

samias *Solve Generalized l_0 Problem with SAMIAS Method*

Description

This function solves the generalized l_0 problem with a sequential list of numbers of knots. The generalized l_0 coefficient is computed via the sequential alternating minimization induced active set (SAMIAS) algorithm. Can deal with any polynomial order or even a general penalty matrix for structural filtering.

Usage

```
samias(y, D = NULL, D_type = c("tf0", "tfq", "user"), q = 0, kmax = min(20,m-1),
     rho = n^(q+1), tmax = 10, eps = 0, smooth = FALSE, h = 5,
     adjust = FALSE, delta = 10, adjust.max = 10, ...)
```

Arguments

`y`	Observed data, of length n.
`D`	Penalty matrix of size m x n (see Details). Default is `NULL`.
`D_type`	Types of D. Either "tf0", "tfq" or "user", depending on what types of contraint that user wants to impose on the data. For `D_type = "tfq"` and `D = NULL`, we solve the l_0 trend filtering of order $q+1$, where q is determined by the argument `q`. For `D_type = "tf0"` and `D = NULL`, we solve the l_0 trend filtering of order 0, i.e., with piecewise constant constraint on the data. For `D_type = "user"`, the penalty matrix `D` must be specified.
`q`	Nonnegative integer used to specify the order of penalty matrix. See `genDtf1d` for Details.
`kmax`	The maximum number of knots. The number of knots is thus $1, 2, \ldots, kmax$. Default is $min(20, m-1)$.
`rho`	The hyperparameter ρ in the augmented Lagrangian of the l_0 trend filtering problem. Default is n^{q+1}.
`tmax`	The maximum number of iterations in the AMIAS algorithm.
`eps`	The tolerance ϵ for an early stoping rule in the SAMIAS algorithm. The early stopping rule is defined as $\|y-\alpha\|_2^2/n \leq \epsilon$, where α is the fitting coefficient.
`smooth`	Whether to smooth the data, if `TRUE`, it smoothes the input data. Default is FALSE.
`h`	Bandwidth in smoothing data. See `my.rollmean` for details.
`adjust`	Whether to adjust the indexes of the active set in the AMIAS algorithm. If `TRUE`, it implements the adjustment when the indexes in the active set are not well separated. Default is FALSE.
`delta`	The minimum gap etween the adjacent knots. Only used when `adjust = TRUE`.
`adjust.max`	The number of iterations in the adjustment step. Only used when `adjust = TRUE`.
`...`	Other arguments.

Details

The generalized l_0 problem with a maximal number of possible knots $kmax$ is

$$\min_{\alpha} 1/2\|y-\alpha\|_2^2 \text{ s.t. } \|D\alpha\|_0 \leq kmax.$$

The penalty matrix D is either the discrete difference operator of order $q+1$ or a general matrix specified by users to enforce some special structure. We solve this problem by sequentially considering the following sub-problem.

$$\min_{\alpha} 1/2\|y-\alpha\|_2^2 \text{ s.t. } \|D\alpha\|_0 = k$$

for k ranging from 1 to $kmax$. This sub-problem can be solved via the AMIAS method, see `amias`. Thus a sequential AMIAS algorithm named SAMIAS is proposed, which is a simple extension of AMIAS by adopting the warm start strategy.

Since it outputs the whole sequence of results for k = 1, . . . , kmax, we can naturally draw the solution path for increasing k and apply the Bayesian information criterion (BIC) for hyperparameter determination. The BIC is defined by

$$nlog(mse) + 2 * log(n) * df,$$

where mse is the mean squared error, i.e., $\|y-\alpha\|_2^2/n$, and $df = k + q + 1$.

Value

A list with class attribute 'samias' and named components:

`call`	The call that produces this object.
`y`	Observe sequence, if smooth, the smooth y will be returned.
`D_type`	Types of D.
`q`	The order of penalty matrix.
`k`	The sequential list of number of knots, i.e., $1, 2, \ldots, kmax$
`alpha`	The optimal coefficients α determined by minimizing BIC.
`v`	The optimal primal variable or split variable of the argumented lagrange form in $D\alpha$.
`u`	The optimal dual variable or lagrange operator of the argumented lagrange form in $D\alpha$ for linear item.
`A`	The optimal estimate of active sets, i.e., set of detected knots.
`df`	Degree of freedom for all the candidate models.
`alpha.all`	Coefficients Matrix with k ranging from 1 to `kmax`, of dimension $nxkmax$.
`v.all`	Matrix of primal variables with k ranging from 1 to `kmax`, of dimension $(n-q-1)xkmax$.
`u.all`	Matrix of dual variables with k ranging from 1 to `kmax`, of dimension $(n-q-1)xkmax$.
`A.all`	List of active sets, of length `kmax`.
`bic`	The BIC value for each value of k.
`iter`	A vector of length `kmax`, record the iterations in the AMIAS algorithm.
`smooth`	Whether to smooth the data.

Author(s)

Canhong Wen, Xueqin Wang, Shijie Quan, Zelin Hong and Aijun Zhang.

Maintainer: Canhong Wen <wench@ustc.edu.cn>

References

Wen, C., Zhu, J., Wang, X., and Zhang, A. (2019) *L0 trend filtering*, technique report.

See Also

print.samias, coef.samias and plot.samias method, and the amias function.

Examples

```
##----- A toy example of piecewise constant signal -------
set.seed(0)
n <- 100
x = seq(1/n, 1,length.out = n)
y0 = 0*x; y0[x>0.5] = 1
y = y0 + rnorm(n, sd = 0.1)
fit <- samias(y, kmax = 5)
op <- par(mfrow=c(1,2))
plot(fit, type= "coef", add.knots = FALSE, main = "Piecewise Constant")
plot(fit, type = "vpath", main = "Piecewise Constant")
par(op)


##----- A toy example of piecewise linear trend -------
set.seed(0)
y0 = 2*(0.5-x); y0[x>0.5] = 2*(x[x>0.5]-0.5)
y = y0 + rnorm(n, sd = 0.1)
fit <- samias(y, D_type = "tfq", q = 1, kmax = 5)
print(fit)


##------ Piecewise constant trend filtering example in Wen et al.(2018).-----
set.seed(1)
data <- SimuBlocks(2048)
fit <- samias(data$y, kmax = 15)           # With default input argument
plot(fit, type="coef", main = "Blocks")    # Plot the optimal estimate
lines(data$x, data$y0, type="s")           # Add the true signal for reference
plot(fit, type= "vpath", main = "Blocks")  # Plot the solution path


##------ Piecewise linear trend filtering example in Wen et al.(2018).-----
set.seed(1)
data <- SimuWave(512)

# samias With adjustment
fit <- samias(data$y, q = 1, D_type = "tfq", kmax = 15, adjust = TRUE, delta = 20)
plot(fit, main = "Wave", k = 10)          # Plot the estimate with user-specified k
```

```
lines(data$x, data$y0, type="l")          # Add the true signal for reference
plot(fit, type= "vpath", main = "Wave")   # Plot the solution path


##------ Doppler example in Wen et al.(2018).-----
set.seed(1)
data <- SimuDoppler(1024)
op <- par(mfrow=c(1,2))
fit1 <- samias(data$y, q = 2, D_type = "tfq", kmax = 30) # piecewise quadratic
plot(fit1, main = "Doppler: q = 2")
fit2 <- samias(data$y, q = 3, D_type = "tfq", kmax = 30) # piecewise Cubic polynomial
plot(fit2, main = "Doppler: q = 3")
par(op)
```

SimuBlocks *Generate simulated data*

Description

Generate data for simulations.

Usage

```
SimuBlocks(n, sigma = 0.1, seed = NA)
SimuWave(n, sigma = 0.1, seed = NA)
SimuDoppler(n, sigma = 0.1, seed = NA)
```

Arguments

n	The number of observations.
sigma	A positive value to specify the standard deviation of the Gaussian noise.
seed	a single value to specify the seed used for generating data. See set.seed for details.

Details

Let $x = i/n, i = 1, \ldots, n$. Assume that y_x follows an underlying trend $y0_x$ with additive noise ϵ_x, i.e.,

$$y_x = y0_x + \epsilon_x,$$

where $y0_x$ is a smooth curve representing relatively long-term movements and ϵ_x is often assumed to follow i.i.d. $N(0, \sigma^2)$.

Value

A list with the following components:

y	The simulated data of length n.
x	The locations of y, of length n.
y0	The true signal of length n.
tau	Locations of the true knots.
SetA	Indexs of the true knots.

Author(s)

Canhong Wen, Xueqin Wang, Shijie Quan, Zelin Hong and Aijun Zhang.

Maintainer: Canhong Wen <wench@ustc.edu.cn>

References

Wen, C., Zhu, J., Wang, X., and Zhang, A. (2019) *L0 trend filtering*, technique report.

Examples

```
##------ Piecewise constant trend filtering example in Wen et al.(2018).-----
set.seed(1)
data <- SimuBlocks(2048)
fit <- samias(data$y, kmax = 15)           # With default input argument
plot(fit, type="coef", main = "Blocks")    # Plot the optimal estimate
lines(data$x, data$y0, type="s")           # Add the true signal for reference
plot(fit, type= "vpath", main = "Blocks")  # Plot the solution path
```

Index