Package: FastSF (via r-universe)

October 16, 2024

Type Package

Title Fast Structural Filtering

Version 0.1.1

Author Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

Maintainer Canhong Wen <wencanhong@gmail.com>

Description An implementation of the fast structural filtering with L0 penalty. It includes an adaptive polynomial estimator by minimizing the least squares error with constraints on the number of breaks in their (k + 1)-st discrete derivative, for a chosen integer k >= 0. It also includes generalized structure sparsity constraint, i.e., graph trend filtering. This package is implemented via the primal dual active set algorithm, which formulates estimates and residuals as primal and dual variables, and utilizes efficient active set selection strategies based on the properties of the primal and dual variables.

License GPL-3

Encoding UTF-8

Depends R (>= 3.0.0)

LazyData true

Imports Rcpp (>= 0.12.10), limSolve

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-07-19 07:32:21 UTC

Contents

`ffused` *Fast Fused Regression*

Description

This is a fast calculation function that solves the L0 fused problem via the primal dual active set algorithm. It fits a piecewise constant regression model by minimizing the least squares error with constraints on the number of breaks in their 1-st discrete derivative.

Usage

```
ffused(y, s, K.max=5)
```

Arguments

`y`	Response sequence to be fitted.
`s`	Number of knots in the piecewise constant(breaks in the derivative), default is 10.
`K.max`	The maximum number of steps for the algorithm to take before termination. Default is 5.

Value

`y`	The observed response vector. Useful for plotting and other methods.
`beta`	Fitted value.
`v`	Primal coefficient. The indexes of the nonzero values correspond to the locations of the breaks.

Author(s)

Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

References

Wen,C., Wang, X., Shen, Y., and Zhang, A. (2017). "L0 trend filtering", technical report.

See Also

plotl0.

Examples

```
set.seed(111)
n <- 1000
sigma <- 0.5
y0 <- rep(0,n)
y0[100:150] <- 2.5
y0[400:600] <- -2.4
y0[800:810] <- 4
y <- y0 + sigma*rnorm(n)

re = ffused(y, s = 8, K.max = 5)
```

---

ffused.ada — *Adaptive Fast Fused Regression*

Description

This is a fast calculation function that solves the L0 fused problem via the primal dual active set algorithm. It fits a piecewise constant regression model by minimizing the number of breaks in derivative with constraints on the least squares error.

Usage

```
ffused.ada(y, tau = 1, s.max = 20, eps = 0.1)
```

Arguments

y	Numeric vector of inputs.
tau	Step length for searching the best model, i.e., in the t-th iteration, a model with tau*t knots will be fitted.
s.max	The maximum nubmer of knots in the piecewise constant(breaks in the (k+1)-st derivative), default is 20
eps	Early stop criterion. The algorithm stops when mean squared error is less than eps

Value

y	The observed response vector. Useful for plotting and other methods.
beta	Fitted value
v	Primal coefficient. The indexes of the nonzero values correspond to the locations of the breaks.

beta.all	Solution path of fitted value, beta, corresponding to different degrees of freedom.
df	A vector giving an unbiased estimate of the degrees of freedom of the fit, i.e., the number of nonzero values in v.

Author(s)

Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

References

Wen,C., Wang, X., Shen, Y., and Zhang, A. (2017). "L0 trend filtering", technical report.

See Also

ffused.

Examples

```
set.seed(111)
n <- 1000
sigma <- 0.5
y0 <- rep(0,n)
y0[100:150] <- 2.5
y0[400:600] <- -2.4
y0[800:810] <- 4
y <- y0 + sigma*rnorm(n)


re = ffused.ada(y, tau = 1, s.max = 10)
```

---

fsf *Fast Structural Filtering*

Description

This function solves the generalized structural filtering problem via the primal dual active set algorithm. It fits a non-parametric regression model by minimizing the least squares error with penalty matrix D on coefficient beta.

Usage

```
fsf(y, D, s = 20, K.max = 5, ddinv=NULL)
```

Arguments

- `y` Response sequence to be filtered.
- `D` Penalty matrix on coeffient beta.
- `s` Number of knots in the penalized coefficient(breaks in the `D*beta`), default is 20.
- `K.max` The maximum number of steps for the algorithm to take before termination. Default is 5.
- `ddinv` The inverse matrix of `D*t(D)`, could be NULL input.

Value

- `y` The observed response vector. Useful for plotting and other methods.
- `beta` Fitted value.
- `v` Primal coefficient. The indexes of the nonzero values correspond to the locations of the breaks in `D*beta`.

Author(s)

Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

References

Wen,C., Wang, X., Shen, Y., and Zhang, A. (2017). "L0 trend filtering", technical report.

See Also

`plotl0`.

Examples

```
require(limSolve)
n <- 1000
sigma <- 0.5
y0 <- rep(0,n)
y0[100:150] <- 2
y0[400:600] <- -1
y0[800:810] <- 4
y <- y0 + sigma*rnorm(n)
y[800:810] <- y0[800:810] + sigma*rnorm(11)
D0 <- matrix(0, n-1,n)
diag(D0) <- -1
for(i in 1:(n-1)) D0[i,i+1] <- 1
m <- dim(D0)[1]

re = fsf(y, D0)
```

fsf.ada *Adaptive Fast Structural Filtering*

Description

This is a function that solves the structural filtering problem with L0 penalty via the primal dual active set algorithm. It fits a non-parametric regression model by minimizing the number of nonzero values in D*beta with constraints on the least squares error.

Usage

```
fsf.ada(y, D, tau=1, s.max=20,  eps=0.1, ddinv=NULL)
```

Arguments

`y`	Response sequence to be filtered.
`D`	Penalty matrix on coeffient beta.
`tau`	Step length for searching the best model, i.e., in the t-th iteration, a model with tau*t knots will be fitted.
`s.max`	The maximum nubmer of knots in the penalized coefficient(breaks in `D*beta`, default is 20
`eps`	Early stop criterion. The algorithm stops when mean squared error is less than eps.
`ddinv`	The inverse matrix of `D*t(D)`, could be NULL input.

Value

`y`	The observed response vector. Useful for plotting and other methods.
`beta`	Fitted value
`v`	Primal coefficient. The indexes of the nonzero values correspond to the locations of the breaks.
`beta.all`	Solution path of fitted value, beta, corresponding to different degrees of freedom.
`df`	A vector giving an unbiased estimate of the degrees of freedom of the fit, i.e., the number of nonzero values in `v`.

Author(s)

Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

References

Wen,C., Wang, X., Shen, Y., and Zhang, A. (2017). "L0 trend filtering", technical report.

See Also

`plotl0`.

Examples

```
require(limSolve)
n <- 1000
sigma <- 0.5
y0 <- rep(0,n)
y0[100:150] <- 2
y0[400:600] <- -1
y0[800:810] <- 4
y <- y0 + sigma*rnorm(n)
y[800:810] <- y0[800:810] + sigma*rnorm(11)
D0 <- matrix(0, n-1,n)
diag(D0) <- -1
for(i in 1:(n-1)) D0[i,i+1] <- 1
ddt <- D0%*%t(D0)
ddinv<- Solve.banded(ddt, 1,1, B = diag(1,dim(D0)[1]))

re <- fsf.ada(y, D0, tau = 1, s.max = 10, eps = 0.1, ddinv = ddinv)
```

---

fsfused *Fast Sparse Fused Regression*

Description

This is a function that solves the L0 fused problem via the primal dual active set algorithm in sparse condition. It fits a piecewise constant regression model by minimizing the least squares error with constraints on the number of breaks in their discrete derivative.

Usage

```
fsfused(y, s = 10, T, K.max=5)
```

Arguments

y	Response sequence to be fitted.
s	Number of knots in the piecewise constant(breaks in the derivative), default is 10.
T	Number of non-zero values in fitted coefficient.
K.max	The maximum number of steps for the algorithm to take before termination. Default is 5.

Value

y	The observed response vector. Useful for plotting and other methods.
beta	Fitted value.
v	Primal coefficient. The indexes of the nonzero values correspond to the locations of the breaks.

Author(s)

Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

References

Wen,C., Wang, X., Shen, Y., and Zhang, A. (2017). "L0 trend filtering", technical report.

See Also

plotl0.

Examples

```
n <- 1000
sigma <- 0.5
y0 <- rep(0,n)
y0[100:150] <- 2.5
y0[400:600] <- -2.4
y0[800:810] <- 4
y <- y0 + sigma*rnorm(n)

re = fsfused(y, s = 10, T = 300)
```

---

fsfused.ada *Adaptive Fast Sparse Fused Regression*

Description

This is a function that solves the L0 fused problem via the primal dual active set algorithm in sparse condition. It fits a piecewise constant regression model by minimizing the number of breaks in derivative with constraints on the least squares error.

Usage

```
fsfused.ada(y, tau=1, s.max=20, T, eps=0.1)
```

Arguments

y	Numeric vector of inputs.
tau	Step length for searching the best model, i.e., in the t-th iteration, a model with tau*t knots will be fitted.
s.max	The maximum nubmer of knots in the piecewise constant(breaks in the (k+1)-st derivative), default is 20
T	Number of non-zero values in fitted coefficient.
eps	Early stop criterion. The algorithm stops when mean squared error is less than eps

Value

y	The observed response vector. Useful for plotting and other methods.
beta	Fitted value
v	Primal coefficient. The indexes of the nonzero values correspond to the locations of the breaks.
beta.all	Solution path of fitted value, beta, corresponding to different degrees of freedom.
df	A vector giving an unbiased estimate of the degrees of freedom of the fit, i.e., the number of nonzero values in v.

Author(s)

Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

References

Wen,C., Wang, X., Shen, Y., and Zhang, A. (2017). "L0 trend filtering", technical report.

See Also

fsfused.

Examples

```
n <- 1000
sigma <- 0.5
y0 <- rep(0,n)
y0[100:150] <- 2.5
y0[400:600] <- -2.4
y0[800:810] <- 4
y <- y0 + sigma*rnorm(n)

re = fsfused.ada(y, tau=1, s.max=10, T = 260, eps=1.2*sigma^2)
```

---

ftf *Fast Trend Filtering*

Description

This function solves the structural filtering problem via the primal dual active set algorithm. It fits a k-th order piecewise polynomial by minimizing the least squares error with constraints on the number of breaks in their (k + 1)-st discrete derivative, for a chosen integer k >= 0.

Usage

```
ftf(y, k = 1, s = 20, K.max = 5)
```

Arguments

`y`	Numeric vector of inputs.
`k`	An integer specifying the desired order of the piecewise polyomial produced by the solution of the trend filtering problem. Must be non-negative, and the default to 1 (linear trend filtering).
`s`	Number of knots in the piecewise polynomial(breaks in the (k+1)-st derivative), default is 20.
`K.max`	The maximum number of steps for the algorithm to take before termination. Default is 5.

Details

The L0 trend filtering fits an adaptive piecewise polynomial to linearly ordered observations with contraints on the number of knots, for a chosen integer k >= 0. The knots or the breaks in their (k + 1)-st discrete derivative are chosen adaptively based on the observations.

Value

`y`	The observed response vector. Useful for plotting and other methods.
`beta`	Filtered value.
`v`	Primal coefficient. The indexes of the nonzero values correspond to the locations of the breaks.

Author(s)

Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

References

Wen,C., Wang, X., Shen, Y., and Zhang, A. (2017). "L0 trend filtering", technical report.

See Also

`plotl0`.

Examples

```
set.seed(1)

sigma <- 0.5
y0 <- c((10:30)/3, (40:10)/4, 2:8)
y <- y0 + sigma*rnorm(length(y0))
re <- ftf(y, k = 1, s = 5)
```

ftf.ada *Adaptive Fast Trend Filtering*

Description

This is a function that adaptively solves the trend filtering problem with L0 penalty via the primal dual active set algorithm. It fits a k-th order piecewise polynomial by minimizing the number of breaks in the (k + 1)-st discrete derivative with the constraints on the least squares error.

Usage

```
ftf.ada(y, k = 1, tau = 1, s.max=20, eps=0.1)
```

Arguments

`y`	Numeric vector of inputs.
`k`	An integer specifying the desired order of the piecewise polyomial produced by the solution of the trend filtering problem. Must be non-negative, and the default to 1 (linear trend filtering).
`tau`	Step length for searching the best model, i.e., in the t-th iteration, a model with tau*t knots will be fitted.
`s.max`	The maximum nubmer of knots in the piecewise polynomial(breaks in the (k+1)-st derivative), default is 20
`eps`	Early stop criterion. The algorithm stops when mean squared error is less than eps

Details

The L0 trend filtering fits an adaptive piecewise polynomial to linearly ordered observations with contraints on the number of knots, for a chosen integer k >= 0. The knots or the breaks in their (k + 1)-st discrete derivative are chosen adaptively based on the observations.

Value

`y`	The observed response vector. Useful for plotting and other methods.
`beta`	Filtered value
`v`	Primal coefficient. The indexes of the nonzero values correspond to the locations of the breaks.
`beta.all`	Solution path of filtered value, beta, corresponding to different degrees of freedom.
`df`	A vector giving an unbiased estimate of the degrees of freedom of the fit, i.e., the number of nonzero values in `v`.

Author(s)

Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

References

Wen,C., Wang, X., Shen, Y., and Zhang, A. (2017). "L0 trend filtering", technical report.

See Also

ftf.

Examples

```
set.seed(1)

sigma <- 0.5
y0 <- c((10:30)/3, (40:10)/4, 2:8)
y <- y0 + sigma*rnorm(length(y0))
re <- ftf.ada(y, k = 1, s.max = 5)
```

---

l0fused_c *L0 Fused Regression*

Description

This is a cpp function used for R function l0fused.

Usage

```
l0fused_c(y, T0, max_steps)
```

Arguments

y	Response sequence to be fitted.
T0	Number of knots in the piecewise constant(breaks in the derivative), i.e., the same as s.
max_steps	The maximum number of steps for the algorithm to take before termination, i.e., the same as K.max.

Value

beta	Fitted value.
u	Dual coefficient.
z	Primal coefficient.

Author(s)

Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

References

Wen,C., Wang, X., Shen, Y., and Zhang, A. (2017). "L0 trend filtering", technical report.

---

l0gen_c *L0 Generalized Regression*

Description

This is a cpp function used for R function l0gen.

Usage

```
l0gen_c(y, D, T0, max_steps, ddinv)
```

Arguments

y	Response sequence to be filtered.
D	Penalty matrix on coeffient beta.
T0	Number of knots in the penalized coefficient(breaks in the D*beta), same as s.
max_steps	The maximum number of steps for the algorithm to take before termination. Same as K.max.
ddinv	The inverse matrix of D*t(D), could be NULL input.

Value

beta	Fitted value.
u	Dual coefficient.
z	Primal coefficient.

Author(s)

Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

References

Wen,C., Wang, X., Shen, Y., and Zhang, A. (2017). "L0 trend filtering", technical report.

l0tf_c *L0 Trend Filtering*

Description

This is a cpp function used for R function l0tf.

Usage

```
l0tf_c(y, k0, T0, max_steps)
```

Arguments

y	Numeric vector of inputs.
k0	An integer specifying the desired order of the piecewise polyomial produced by the solution of the trend filtering problem.
T0	Number of knots in the piecewise polynomial(breaks in the (k+1)-st derivative).
max_steps	The maximum number of steps for the algorithm to take before termination.

Value

beta	Fitted value.
u	Dual coefficient.
z	Primal coefficient.

Author(s)

Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

References

Wen,C., Wang, X., Shen, Y., and Zhang, A. (2017). "L0 trend filtering", technical report.

plotl0 *Plot L0 fitted value*

Description

This is a function that plots the fitted results of L0 method.

Usage

```
plotl0(re)
```

Arguments

re	The list generated by functions in FastSF package. For a given degree of freedom, plot the fitted value of L0 method. For adaptive situations, plot the MSE of every model under different degrees of freedom.

Author(s)

Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

---

sl0fused_c *Sparse L0 Fused Regression*

Description

This is a cpp function used for R function sl0fused.

Usage

```
sl0fused_c(y, T0, T02, max_steps)
```

Arguments

y	Response sequence to be fitted.
T0	Number of knots in the piecewise constant(breaks in the derivative).
T02	Number of non-zero values in fitted coefficient.
max_steps	The maximum number of steps for the algorithm to take before termination.

Value

beta	Fitted value.
u	Dual coefficient.
z	Primal coefficient.
d	Dual coefficient.

Author(s)

Canhong Wen, Xueqin Wang, Yanhe Shen, Aijun Zhang

References

Wen,C., Wang, X., Shen, Y., and Zhang, A. (2017). "L0 trend filtering", technical report.

Index