Package 'moseg'

June 15, 2023

Type Package

Title Data segmentation for sparse regression models

Version 0.1.0

Imports glmnet, MASS, doParallel, foreach, parallel, Matrix

Maintainer Dom Owens <dom.owens@bristol.ac.uk>

Description
Detects and locates change points in the sparse regression model, based on Lasso estimation and using moving sum (MOSUM) statistics.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Depends R (>= 3.5.0)

R topics documented:

`eqdata` *Equity premium and macro/financial variables*

Description

A dataset containing time series of the equity premium and other macro/financial variables

Usage

```
eqdata
```

Format

A data frame with 1032 rows and 16 variables:

yyyymm Date

b.m Book-to-Market Ratio

tbl Treasury Bills

lty Long Term Yield

ntis Net Equity Expansion

infl Inflation

ltr Long Term Rate of Returns

svar Stock Variance

equity_premium Equity Premium

dp Dividend Price Ratio

dy Dividend Yield

ep Earnings Price Ratio

de Dividend Payout Ratio

tms Term Spread

dfy Default Yield Spread

dfr Default Return Spread

Source

https://sites.google.com/view/agoyal145/?redirpath=/

References

Welch, Ivo, and Amit Goyal. "A comprehensive look at the empirical performance of equity premium prediction." The Review of Financial Studies 21.4 (2008): 1455-1508.

moseg *Detect and estimate multiple change points in the sparse regression model*

Description

Detect and estimate multiple change points in the sparse regression model

Usage

```
moseg(
  X,
  y,
  G = NULL,
  lambda = c("min", "1se"),
  family = c("gaussian", "binomial", "poisson"),
  threshold = NULL,
  n.cps = NULL,
  grid.resolution = 1/G,
  nu = 0.5,
  do.refinement = TRUE,
  do.plot = TRUE,
  do.scale = TRUE,
  ncores = 1,
  ...
)
```

Arguments

X	design matrix
y	response vector
G	integer bandwidth; default is chosen based on dimensions of X
lambda	regularisation parameter; either a numeric, or one of "min","1se" (see cv.glmnet)
family	response type, one of "gaussian","binomial","poisson"
threshold	numeric test rejection threshold; see details for default
n.cps	chosen number of change points to return; if specified, overrides threshold
grid.resolution	controls number of subsamples to take
nu	numeric localisation tuning parameter
do.refinement	Boolean - perform location refinement
do.plot	Boolean - return plots
do.scale	Boolean - centre and scale X,y
ncores	number of parallel cores
...	optional arguments to glmnet

Details

The default threshold is chosen as a product of exponents of `lambda` and `G`. For a more accurate, but slightly slower, procedure, see moseg.cv.

Value

List containing

- `mosum` numeric vector of mosum detector
- `cps` integer vector of estimated change points
- `refined.cps` integer vector of refined change points
- `G` input
- `lambda` input
- `threshold` input
- `grid.resolution` input
- `family` input
- `plots` list of detector and refinement plots
- `model_list` list of fitted `glmnet` models at each grid point

Examples

```
eqX <- eqdata[,-c(1,9)]
eq_mosum <- moseg(as.matrix(eqX), eqdata[,9], 120, grid.resolution = 1/10, ncores = 2)
```

`moseg.cv` *Detect and estimate multiple change points in the sparse regression model, selecting the number of change points using sample splitting*

Description

Detect and estimate multiple change points in the sparse regression model, selecting the number of change points using sample splitting

Usage

```
moseg.cv(
  X,
  y,
  G = NULL,
  lambda = NULL,
  max.cps = NULL,
  family = c("gaussian", "binomial", "poisson"),
  loss = c("1", "2"),
  folds = 2,
  path.length = 5,
```

```
  grid.resolution = 1/G,
  nu = 0.5,
  do.plot = TRUE,
  do.scale = TRUE,
  do.refinement = TRUE,
  ncores = 1,
  ...
)
```

Arguments

X	design matrix
y	response vector
G	integer bandwidth; default is chosen based on dimensions of X
lambda	vector of numeric regularisation parameters
max.cps	maximum number of change points to consider
family	response type, one of "gaussian","binomial","poisson"
loss	l-norm for CV loss function, one of "1", "2"
folds	number of folds for CV
path.length	number of lambda values to consider
grid.resolution	controls number of subsamples to take
nu	numeric localisation tuning parameter
do.plot	Boolean - return plots
do.scale	Boolean - centre and scale X,y
do.refinement	Boolean - perform location refinement
ncores	number of parallel cores
...	optional arguments to glmnet

Value

moseg.cv object containing

- mosum numeric vector of mosum detector
- cps integer vector of estimated change points
- refined.cps integer vector of refined change points
- G input
- lambda selected regularisation parameter
- threshold implied threshold
- detectors mosum detector series for each lambda value
- cv matrix of cross-validation errors
- model_list list of fitted piecewise models
- plots list of detector and refinement plots
- family input

Examples

```
eqX <- eqdata[,-c(1,9)]
eq_thr <- moseg.cv(as.matrix(eqX), as.matrix(eqdata[,9]), G=120, max.cps = 3, ncores = 2)
```

`moseg.fit` *Fit a piecewise sparse regression model and evaluates a penalty. Fits a model to each stationary segment using* `glmnet::cv.glmnet`.

Description

Fit a piecewise sparse regression model and evaluates a penalty. Fits a model to each stationary segment using `glmnet::cv.glmnet`.

Usage

```
moseg.fit(
  X,
  y,
  cps,
  lambda = c("min", "1se"),
  family = c("gaussian", "binomial", "poisson"),
  type = c("link", "response", "coefficients", "nonzero", "class"),
  do.plot = TRUE,
  do.scale = TRUE,
  ...
)
```

Arguments

`X`	design matrix
`y`	response vector
`cps`	vector of change point locations
`lambda`	regularisation parameter; either a numeric, or one of `"min"`,`"1se"` (see cv.glmnet)
`family`	response type, one of `"gaussian"`,`"binomial"`,`"poisson"`
`type`	type of prediction; see `?glmnet.predict`
`do.plot`	Boolean - return coefficent plot
`do.scale`	Boolean - scale `X`,`y`
`...`	optional arguments to `glmnet`

Value

list containing

- `model` list of fitted models
- `likelihood` likelihood value
- `preds` vector of fitted values for each time step
- `coeffs` coefficient matrix
- `plot` coefficient heatmap

Examples

```
eqX <- eqdata[,-c(1,9)]
eq_mosum <- moseg.fit(as.matrix(eqX), as.matrix(eqdata[,9]), c(500))
```

`moseg.ms` *Detect and estimate multiple change points in the sparse regression model using multiple bandwidths*

Description

Detect and estimate multiple change points in the sparse regression model using multiple bandwidths

Usage

```
moseg.ms(
  X,
  y,
  Gset,
  lambda = c("min", "1se"),
  family = c("gaussian", "binomial", "poisson"),
  threshold = NULL,
  grid.resolution = NULL,
  nu = 0.5,
  do.plot = TRUE,
  do.scale = TRUE,
  ncores = 1,
  ...
)
```

Arguments

`X`	design matrix
`y`	response vector
`Gset`	integer vector of bandwidths; default is chosen based on dimensions of `X`

lambda	regularisation parameter; either a numeric, or one of "min","1se" (see cv.glmnet)
family	response type, one of "gaussian","binomial","poisson"
threshold	numeric test rejection threshold; see moseg for default choice
grid.resolution	controls number of subsamples to take
nu	numeric localisation tuning parameter
do.plot	Boolean - return plots
do.scale	Boolean - scale X,y
ncores	number of parallel cores
...	optional arguments to glmnet

Value

List containing

- cps integer vector of estimated change points
- plot multiscale plot
- moseg.G list of 'moseg' objects corresponding to 'Gset' in ascending order

See Also

moseg.ms.cv

Examples

```
eqX <- eqdata[,-c(1,9)]
eq_mosum <- moseg.ms(as.matrix(eqX), eqdata[,9], c(60,90,120), ncores = 2)
```

`moseg.ms.cv` *Detect and estimate multiple change points in the sparse regression model using multiple bandwidths, selecting the number of change points using sample splitting*

Description

Detect and estimate multiple change points in the sparse regression model using multiple bandwidths, selecting the number of change points using sample splitting

Usage

```
moseg.ms.cv(
  X,
  y,
  Gset = NULL,
  lambda = NULL,
  family = c("gaussian", "binomial", "poisson"),
  loss = c("1", "2"),
  folds = 1,
  path.length = 5,
  threshold = NULL,
  grid.resolution = 1/Gset,
  nu = 0.5,
  do.plot = TRUE,
  do.scale = TRUE,
  ncores = 1,
  ...
)
```

Arguments

X	design matrix
y	response vector
Gset	integer vector of bandwidths; default is chosen based on dimensions of X
lambda	regularisation parameter; either a numeric, or one of "min","1se" (see cv.glmnet)
family	response type, one of "gaussian","binomial","poisson"
loss	l-norm for CV loss function, one of "1", "2"
folds	number of folds for CV
path.length	number of lambda values to consider
threshold	numeric test rejection threshold; see reference for default choice
grid.resolution	controls number of subsamples to take
nu	numeric localisation tuning parameter
do.plot	Boolean - return plots
do.scale	Boolean - scale X,y
ncores	number of parallel cores
...	optional arguments to glmnet

Value

moseg.ms.cv object containing

- anchors integer vector of estimated change points
- refined.cps integer vector of refined change points
- moseg.G list of ‘moseg.cv‘ objects corresponding to ‘Gset‘ in ascending order

Examples

```
eqX <- eqdata[,-c(1,9)]
eq_mosum <- moseg.ms.cv(as.matrix(eqX), eqdata[,9], c(60,90,120), ncores = 2)
```

`moseg.sim` *Simulate from a sparse piecewise regression model*

Description

Simulate from a sparse piecewise regression model

Usage

```
moseg.sim(
  n,
  p,
  sparsity = floor(sqrt(p)),
  q = 1,
  sigma.noise = 1,
  sigma.x = c("id", "band", "ar"),
  kappa = 1
)
```

Arguments

`n`	sample size
`p`	number of parameters
`sparsity`	number of non-zero parameters
`q`	number of change points
`sigma.noise`	error standard deviation
`sigma.x`	covariance structure of `X`, one of `"id"`, `"band"`, `"ar"`
`kappa`	change size

Value

List containing

- `y` response vector
- `X` matrix of covariates
- `cps` vector of change points
- `beta` matrix of parameters corresponding to each regime

and all inputs

Examples

```
moseg.sim(100, 50)
```

plot.moseg *Plot the moseg detector*

Description

Plotting method for S3 objects of class `moseg`.

Usage

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

Arguments

`x`	`moseg` object
`...`	unused

Value

A detector plot

plot.moseg.cv *Plot the moseg detector*

Description

Plotting method for S3 objects of class `moseg.cv`.

Usage

```
## S3 method for class 'moseg.cv'
plot(x, type = c("cv", "mosum"), ...)
```

Arguments

`x`	`moseg.cv` object
`type`	plot type, one of `"cv"`, `"mosum"`
`...`	unused

Value

A cv or detector plot

plot.moseg.ms *Plot the multiscale moseg detector*

Description

Plotting method for S3 objects of class `moseg.ms`.

Usage

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

Arguments

`x`	`moseg.ms` object
`...`	unused

Value

Detector plots

plot.moseg.ms.cv *Plot the multiscale moseg.cv detector*

Description

Plotting method for S3 objects of class `moseg.ms.cv`.

Usage

```
## S3 method for class 'moseg.ms.cv'
plot(x, type = c("cv", "mosum"), ...)
```

Arguments

`x`	`moseg.ms` object
`type`	plot type, one of `"cv"`, `"mosum"`
`...`	unused

Value

cv or detector plots

Index