Package 'mosumsr'

June 17, 2022

Type Package

Title Data segmentation for sparse regression models

Version 0.1.0

Imports glmnet, MASS, doParallel, foreach, parallel

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.

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

Description

Detect and estimate multiple change points in the sparse regression model

Usage

```
mosumsr(
  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 = NULL,
  ...
)
```

Arguments

`X`	design matrix
`y`	response vector
`G`	integer bandwidth; defaults to `round(30 + ncol(X)/100)`
`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 mosumsr.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
- `threshold`
- `lambda`
- `model_list` list of fitted `glmnet` models at each grid point
- `plots` list of detector and refinement plots
- `family` input

Examples

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

`mosumsr.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

```
mosumsr.cv(
  X,
  y,
  G = NULL,
  lambda = NULL,
  max.cps = NULL,
  family = c("gaussian", "binomial", "poisson"),
  path.length = 5,
  grid.resolution = 1/G,
  nu = 0.5,
  do.plot = TRUE,
  do.scale = TRUE,
```

```
  do.refinement = TRUE,
  ncores = NULL,
  ...
)
```

Arguments

X	design matrix
y	response vector
G	integer bandwidth; defaults to round(30 + ncol(X)/100)
lambda	vector of numeric regularisation parameters
max.cps	maximum number of change points to consider
family	response type, one of "gaussian","binomial","poisson"
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

List containing

- mosum numeric vector of mosum detector
- cps integer vector of estimated change points
- refined_cps integer vector of refined change points
- 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 <- mosumsr.cv(as.matrix(eqX), as.matrix(eqdata[,9]), G=120, max.cps = 3, ncores = 2)
```

`mosumsr.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

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

`mosumsr.multiscale` *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

```
mosumsr.multiscale(
  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 = NULL,
  ...
)
```

Arguments

`X`	design matrix
`y`	response vector
`Gset`	integer vector of bandwidths; default smallest is `round(30 + ncol(X)/100)`
`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 mosumsr 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
- mosumsr.G list of ‘mosumsr‘ objects corresponding to ‘Gset‘ in ascending order

See Also

mosumsr.multiscale.cv

Examples

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

---

mosumsr.multiscale.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

```
mosumsr.multiscale.cv(
  X,
  y,
  Gset = NULL,
  lambda = NULL,
  family = c("gaussian", "binomial", "poisson"),
  threshold = NULL,
  grid.resolution = 1/Gset,
  nu = 0.5,
  do.plot = TRUE,
  do.scale = TRUE,
  ncores = NULL,
  ...
)
```

Arguments

X	design matrix
y	response vector
Gset	integer vector of bandwidths; default smallest is round(30 + ncol(X)/100)
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 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

List containing

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

Examples

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

---

mosumsr.sim *Simulate from a sparse piecewise regression model*

Description

Simulate from a sparse piecewise regression model

Usage

```
mosumsr.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

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

Index