Package ‘TSMCP’

November 18, 2018

Type Package

Title Fast Two Stage Multiple Change Point Detection

Version 1.0

Author Yaguang Li [aut, cre],
Baisuo Jin [aut]

Maintainer Yaguang Li <liyg@mail.ustc.edu.cn>

Depends R (>= 3.3.0), plus, lars

Suggests MASS

Description A novel and fast two stage method for simultaneous multiple change point detection and variable selection for piecewise stationary autoregressive (PSAR) processes and linear regression model. It also simultaneously performs variable selection for each autoregressive model and hence the order selection.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2018-11-18 19:00:09 UTC

R topics documented:

`cpvnts` *Multiple change point detection and variable selection for nonstationary time series models.*

Description

This function provides a novel and fast methodology for simultaneous multiple structural break estimation and variable selection for nonstationary time series models by using ASCAD or AMCP to estimate the chang points proposed by Jin, Shi and Wu (2011).

Usage

```
cpvnts(Y, method = c("mcp", "scad"), p, lam.d)
```

Arguments

`Y`	an autoregressive time series.
`method`	the method to be used by mcp or scad. See `plus` in R packages **plus** for details.
`p`	an upper bound of autoregressive orders.
`lam.d`	an parameter of the refining procedure. Suggest lam.d=0.05 if the length of Y is smaller than 5000 and lam.d=0.01 otherwise.

Value

pluscpv returns an object of class "cpvnts". An object of class "cpvnts" is a list containing the following components:

`variable.selection`
: a matrix of variable selection. Element of the matrix is 1 if the variable is selected and 0 otherwise.

`change.points`
: estimators of change points.

References

Jin, B., Shi, X., and Wu, Y. (2013). A novel and fast methodology for simultaneous multiple structural break estimation and variable selection for nonstationary time series models. *Statistics and Computing*, 23(2), 221-231.

See Also

plus

Examples

```
### Example 1: No change point with sample size 1000###

n <- 1000
Y <- rnorm(n)
cp1 <- c(3, 1000)
a0 <- c(0.7)
a1 <- c(1.3)
a2 <- c(-0.8)

for (j in 2:2) {
  for (i in (cp1[j-1] + 1):cp1[j]) {
    Y[i] <- a0[j-1] + a1[j-1] * Y[i-1] + a2[j-1] * Y[i-2] + rnorm(1)
  }
}

ts.plot(Y)

# Y(t)=0.7+1.3Y(t-1)-0.8Y(t-2)+e(t)


mcp.cp <- cpvnts(Y, "mcp", 4, 0.05)
mcp.cp  #result of AMCP

scad.cp <- cpvnts(Y, "scad", 4, 0.05)
scad.cp  #result of ASCAD


### Example 2(Davis et al (2006)):Two change points with sample size
### 1024##
n <- 1024
Y <- rnorm(n)
cp1 <- c(3, 512, 769, 1024)
a1 <- c(0.9, 1.69, 1.32)
a2 <- c(0, -0.81, -0.81)

for (j in 2:4) {
  for (i in (cp1[j - 1] + 1):cp1[j]) {
    Y[i] <- a1[j - 1] * Y[i - 1] + a2[j - 1] * Y[i - 2] + rnorm(1)
  }
}

ts.plot(Y)

# Y(t)=0.9Y(t-1)+e(t),if t<512 Y(t)=1.69Y(t-1)-0.81Y(t-2)+e(t), if
# 512<t<769 Y(t)=1.32Y(t-1)-0.81Y(t-2)+e(t), if 769<t<1024


mcp.cp <- cpvnts(Y, "mcp", 4, 0.05)
mcp.cp  #result of AMCP
```

```
scad.cp <- cpvnts(Y, "scad", 4, 0.05)
scad.cp  #result of ASCAD


### Example 3: Six change points with sample size 10000###
n <- 10000
Y <- rnorm(n)
cp1 <- c(3, 1427, 3084, 4394, 5913, 7422, 8804, 10000)

a1 <- c(1.58, 1.12, 1.61, 1.24, 1.53, 1.32, 1.69)
a2 <- c(-0.79, -0.68, -0.75, -0.66, -0.64, -0.81, -0.81)

for (j in 2:length(cp1)) {
  for (i in (cp1[j - 1] + 1):cp1[j]) {
    Y[i] <- a1[j - 1] * Y[i - 1] + a2[j - 1] * Y[i - 2] + rnorm(1)
  }
}



ts.plot(Y)


mcp.cp <- cpvnts(Y, "mcp", 4, 0.01)
mcp.cp  #result of AMCP

scad.cp <- cpvnts(Y, "scad", 4, 0.01)
scad.cp  #result of ASCAD
```

`tsmcplm` *Two stage multiple change point detection in linear models.*

Description

This function provides a two stage procedure for simultaneously detecting multiple change points in linear models. In the cutting stage, the change point problem is converted into a model selection problem so that a modern model selection method can be applied. In the refining stage, the change points obtained in the cutting stage are finalized via a refining method. The tuning parameter lambda is chosen by BIC.

Usage

```
tsmcplm(Y, X, method = c("lasso", "adapt", "mcp", "scad"), c)
```

Arguments

`Y`	an response vector.

X	the n-by-p design matrix.
method	the method to be used by lasso, adaptive lasso, mcp or scad. See plus in R packages **plus** for details.
c	ceiling(c*sqrt(length(Y))) is the length of each segments in spliting stage.

Value

tsmcplm returns an object of class "tsmcplm". An object of class "tsmcplm" is a list containing the following components:

change.points	estimators of change points.

References

Jin, B., Wu, Y., & Shi, X. (2016). Consistent two-stage multiple change-point detection in linear models. *Canadian Journal of Statistics*, 44(2), 161-179.

See Also

plus lars

Examples

```
## example 1: mean shift model
## true change point location:
## 100, 130, 150, 230, 250, 400, 440, 650, 760, 780, 810
Y <- rnorm(1000, 0, 0.5) +
  c(rep(0,100), rep(4,30),rep(-1,20), rep(2,80), rep(-2,20),
    rep(3,150), rep(-1, 40), rep(1,210), rep(5,110), rep(2,20),
    rep(7,30), rep(3,190))
ts.plot(Y)

##estimate change points
tsmcplm(Y = Y, X = NULL, method = "adapt", c = 0.3)


## example 2: linear model:
## a periodic auto correlation series with period 122 and
## order of auto correlation 1

###
## true change point location:
## 200, 350, 450, 550, 700, and 850


n <- 1000
y <- rnorm(n)
for (t in 2:n) {
  y[t] <- cos(t*pi/61) + 3*sin(t*pi/61) + 0.5*y[t-1] + (2*sin(t*pi/61) +
  0.1 * y[t-1])*(200 < t)+ (2* cos(t*pi/61)- 4 *sin(t*pi/61)- 0.6* y[t-1] )*
  (350 < t) + (2* sin(t*pi/61) + 0.7* y[t-1] )*(450 < t) + (-3* sin(t*pi/61) -
```

```
  0.3* y[t-1] )*(550 < t) + (-3* cos(t*pi/61) + 5* sin(t*pi/61))* (700 < t) +
  (3* cos(t*pi/61) - 5* sin(t*pi/61) - 0.4*y[t-1] )* (850 < t) + rnorm(1)
}
ts.plot(y)

x <- sapply(2:n, function(t){cbind(cos(t*pi/61), sin(t*pi/61), y[t-1])},
            simplify = FALSE)
x <- do.call(rbind, x)
tsmcplm(Y = y[-1], X = x, method = "adapt", c = 2)
```

Index