

Package: rid (via r-universe)

September 5, 2024

Title Multiple Change-Point Detection in Multivariate Time Series

Version 0.0.1

Description Provides efficient functions for detecting multiple change points in multidimensional time series. The models can be piecewise constant or polynomial. Adaptive threshold selection methods are available, see Fan and Wu (2024) [<arXiv:2403.00600>](https://arxiv.org/abs/2403.00600).

License GPL-3

Encoding UTF-8

Depends purrr, CircularSilhouette, pracma

NeedsCompilation yes

RoxygenNote 7.2.1

LinkingTo Rcpp, RcppEigen

Author Xinyuan Fan [aut, cre, cph], Weichi Wu [aut]

Maintainer Xinyuan Fan <fxy22@mails.tsinghua.edu.cn>

Repository CRAN

Date/Publication 2024-03-08 20:50:02 UTC

Contents

List2Matrix	2
localization	2
rid	3

Index	6
--------------	----------

List2Matrix

Convert a list of matrices into a single large matrix

Description

Convert a list of matrices into a single large matrix

Usage

```
List2Matrix(x, sym = FALSE)
```

Arguments

x A numeric list with each entity be a numeric matrix with p rows and q columns

sym A logical scalar representing whether each matrix is symmetric. If true, the duplicated half is removed

Value

A numeric matrix with pq rows and T columns

Examples

```
x=list(x1=1:3,x2=4:6)
List2Matrix(x)

y=list(y1=matrix(1:4,2),y2=matrix(5:8,2))
List2Matrix(y)
```

localization*Localization procedure*

Description

The localization procedure to detect change-points.

Usage

```
localization(data, intervals, l = 0, scaling = FALSE, q = 1)
```

Arguments

data	A numeric matrix of observations with each horizontal axis being time, and each column being the multivariate time series
intervals	A numeric matrix of intervals with each row be a vector representing the interval
l	A non-negative integer of order of polynomial ($l=0$ means piecewise constant)
scaling	A logical scalar representing whether to perform refinement for locally stationary data. Only useful when $l=0$
q	A positive integer of norm

Value

The positions of estimated change-points

Examples

```
set.seed(0)
data=rep(c(0,2,0),each=40)+rnorm(120)
d=rid(data,M=1000,tau="clustering")
cpt=localization(data,d$Good_Intervals)
print(cpt)
```

rid

Random intervals distillation procedure

Description

Distilled intervals that cover change-points are constructed.

Usage

```
rid(
  data,
  M = 1000,
  l = 0,
  scaling = FALSE,
  q = 1,
  intervals = NULL,
  tau = c("clustering", "ref"),
  bw = "nrd0",
  adjust = 0.5,
  k.max = 4,
  adj = 1.3
)
```

Arguments

data	A numeric matrix of observations with each horizontal axis being time, and each column being the multivariate time series
M	A positive integer of random intervals, used only when intervals=NULL
l	A non-negative integer of order of polynomial (l=0 means piecewise constant)
scaling	A logical scalar representing whether to perform refinement for locally stationary data. Only useful when l=0
q	A positive integer of norm
intervals	A numeric matrix of intervals with each row be a vector representing the interval. If intervals=NULL, random intervals are sampled
tau	A non-negative number representing the threshold of detection. If tau="clustering" (only useful when l=0), a clustering-based adaptive approach is applied. If tau="ref", a method based on simulated reference values is applied
bw	A parameter passed into function density . Only useful when tau="clustering"
adjust	A parameter passed into function density . Only useful when tau="clustering"
k.max	A positive integer representing the maximum value of clusters in threshold determining. Only useful when tau="clustering"
adj	A positive number used to multiply onto the threshold tau, providing threshold adjustments for small sample size

Value

A list containing:

Good_Intervals	A numeric matrix with each row being an interval that covers a change-point
Threshold	A positive number of the threshold

See Also

[localization](#)

Examples

```
## An example for the univariate case
set.seed(0)
data=rep(c(0,2,0),each=40)+rnorm(120)
d=rid(data,M=1000,tau="clustering")
cpt=localization(data,d$Good_Intervals)
print(cpt)
```

```
## An example for the multivariate case
set.seed(0)
data1=rep(c(0,2,0),each=40)+rt(120,8)
data2=rep(c(0,2,0),each=40)+rnorm(120)
data=rbind(data1,data2)
d=rid(data,M=1000,tau="clustering")
cpt=localization(data,d$Good_Intervals)
```

```

print(cpt)

## An example for the piecewise polynomial case
set.seed(0)
n=300
cp=c(0,round(n/3),round(2*n/3),n)
mu=matrix(c(0.004,-0.1,0,-0.01,0.02,0,0.01,-0.04,0),nrow=3,byrow = TRUE)
mu1=mu[1,1]*(1:n)^2+mu[1,2]*(1:n)+mu[1,3]
for(j in 2:3){
  index=which((1:n)-cp[j]>0)
  tmp1=(1:n)-cp[j]
  tmp=mu[j,1]*tmp1^2+mu[j,2]*tmp1+mu[j,3]
  tmp[1:(index[1]-1)]=0
  mu1=mu1+tmp
}
data=mu1+runif(n,-6,6)
plot(data,type="l")
d=rid(data,M=500,tau="ref",l=2)
cpt=localization(data,d$Good_Intervals,l=2)
print(cpt)

## An example for refinement in the locally stationary time series
set.seed(0)
n=1000
cp=c(0,round(n/4),round(3*n/4),n)
epsilon=rnorm(500+n,0,1)
ei=rep(0,500+n)
for(j in 2:(500+n)){ei[j]=0.5*ei[j-1]+epsilon[j]}
x=ei[(501):(500+n)]
lrv=purrr::map_dbl(1:n,function(j){sqrt(max(1,2000*j/n))})
x=x*lrv
x=x+c(rep(0,round(n/4)),rep(20,round(n/2)),rep(-20,n-round(n/4)-round(n/2)))
data=x
plot(data,type="l")
d=rid(data,M=1000,scaling = TRUE,tau="clustering")
cpt=localization(data,d$Good_Intervals)
print(cpt)

```

Index

density, [4](#)

List2Matrix, [2](#)

localization, [2](#), [4](#)

rid, [3](#)