Package 'onlineCOV'

October 14, 2022

Type Package

Title Online Change Point Detection in High-Dimensional Covariance Structure

Version 1.3

Date 2020-03-14

Author Lingjun Li and Jun Li

Maintainer Jun Li <jli49@kent.edu>

Description Implement a new stopping rule to detect anomaly in the covariance structure of high-dimensional online data. The detection procedure can be applied to Gaussian or non-Gaussian data with a large number of components. Moreover, it allows both spatial and temporal dependence in data. The dependence can be estimated by a data-driven procedure. The level of threshold in the stopping rule can be determined at a pre-selected average run length. More detail can be seen in Li, L. and Li, J. (2020) ``Online Change-Point Detection in High-Dimensional Covariance Structure with Application to Dynamic Networks." <arXiv:1911.07762>.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-03-23 09:40:02 UTC

R topics documented:

`nuisance.est` *Estimate nuisance parameters in the stopping rule.*

Description

The function estimates the nuisance parameters required in the stopping rule, through a trainig sample.

Usage

```
nuisance.est(training.sample)
```

Arguments

`training.sample`	A historical dataset without change points.

Value

Returns a list of estimated nuisance parameters. See below for more detail.

`mu.hat`	The sample mean of the training sample.
`M.hat`	The estimated M dependence.
`cor.hat`	A value used to obtain the standard deviation of the test statistic in the stopping rule.

References

Li, L. and Li, J. (2020) "Online Change-Point Detection in High-Dimensional Covariance Structure with Application to Dynamic Networks." arXiv:1911.07762.

Examples

```
p<-200;n0<-200
M<-2

Gam1<-diag(1,p,p)

data_Mat<-matrix(0,n0,p)
L<-M+1
Z<-matrix(rnorm(p*(n0+L-1)),p*(n0+L-1),1)
vec.coef<-1/rep(c(L:1),each=p)

  for(j in 1:n0){
  Gam.mat<-t(apply(Gam1,1,rep,L))*matrix(vec.coef,ncol=L*p,nrow=p,byrow=TRUE)
  data_Mat[j,]<-matrix((Gam.mat%*%Z[((j-1)*p+1):((j+L-1)*p),]),1,p,byrow=FALSE)
  }
```

```
training.sample<-data_Mat
nuisance.results<-nuisance.est(training.sample)
mu<-nuisance.results$mu.hat
M<-nuisance.results$M.hat
cor<-nuisance.results$cor.hat
```

`stopping.rule` *Online change-point detection by the stopping rule.*

Description

Function to determine whether a process with continually arrving data should be terminated, based on the proposed stopping rule.

Usage

```
stopping.rule(ARL, H, mu, M, cor, old.data, new.data)
```

Arguments

`ARL`	The expected value of the stopping time when there is no change, eg. ARL = 5000.
`H`	The window size so that the stopping rule only considers H observations from the current time, eg. H=100.
`mu`	The mean vector of the observation with dimension 1 by p, can be estimated from a training sample through the function "nuisance.est".
`M`	M dependence, can be estimated from a training sample through the function "nuisance.est", eg. M=0 means data are temporally independent.
`cor`	A value used to obtain the standard deviation of the test statistic in the stopping rule, can be estimated from a training sample through the function "nuisance.est".
`old.data`	The observed sequence of data. The dataset has dimension H by p, where H is the window size, or the number of observed data (row), and p is the number of components (column).
`new.data`	A newly arrived observation with dimension 1 by p.

Value

Returns a list with items "decision" and "old.updated". See below for more detail.

`decision`	returns 1 if the stopping rule detects a change point, and returns 0 otherwise.
`old.updated`	The updated observed dataset in this step, with dimension H by p. The Hth observation is the newly arrived observation, and the rest H-1 observations come from the previous dataset.

References

Li, L. and Li, J. (2020) "Online Change-Point Detection in High-Dimensional Covariance Structure with Application to Dynamic Networks." arXiv:1911.07762.

Examples

```
# The following is an example to detect covariance structure change
# in a real-time manner, in the sense that we pretend that
# the observations in the dataset continually arrive in time.
# At each time, we determine whether the process should be
# terminated through the proposed stopping rule.
# there is an immediate change point at n0=200


p<-200;n<-10000;n0<-200 #n0 is traing sample size
rho<-0.6;M<-2
H<-100;ARL<-5000

Gam1<-diag(1,p,p)
times<-1:p
d<-abs(outer(times, times, "-"))
sigma<-rho^d
Gam2<-eigen(sigma,symmetric=TRUE)$vectors%*%diag(sqrt(eigen(sigma,symmetric=TRUE)$values),p)
Gam<-cbind(Gam1,Gam2)

data_Mat<-matrix(0,n0,p)
L<-M+1
Z<-matrix(rnorm(p*(n+L-1)),p*(n+L-1),1)
vec.coef<-1/rep(c(L:1),each=p)

  for(j in 1:n0){
  Gam.m<-Gam[,1:p]
  Gam.mat<-t(apply(Gam.m,1,rep,L))*matrix(vec.coef,ncol=L*p,nrow=p,byrow=TRUE)
  data_Mat[j,]<-matrix((Gam.mat%*%Z[((j-1)*p+1):((j+L-1)*p),]),1,p,byrow=FALSE)
  }

old.data<-data_Mat
nuisance.results<-nuisance.est(old.data)
mu<-nuisance.results$mu.hat
M<-nuisance.results$M.hat
cor<-nuisance.results$cor.hat

 j<-n0+1;decision = 0

  while(decision==0){

  Gam.m<-Gam[,(p+1):(2*p)]
  Gam.mat<-t(apply(Gam.m,1,rep,L))*matrix(vec.coef,ncol=L*p,nrow=p,byrow=TRUE)
  new.data<-matrix((Gam.mat%*%Z[((j-1)*p+1):((j+L-1)*p),]),1,p,byrow=FALSE)

  result<-stopping.rule(ARL,H,mu,M,cor,old.data,new.data)
```

```
 decision<-result$decision
 old.data<-result$old.updated
 cpt.est<-j-n0

 j<-j+1
 }

print(cpt.est) #The point where the detection procedure terminates.
```

Index