Package 'OLCPM'

January 20, 2025

Type Package

Title Online Change Point Detection for Matrix-Valued Time Series

Version 0.1.2

Author Yong He [aut],
Xinbing Kong [aut],
Lorenzo Trapani [aut],
Long Yu [aut, cre]

Maintainer Long Yu <fduyulong@163.com>

Description We provide two algorithms for monitoring change points with online matrix-valued time series, under the assumption of a two-way factor structure. The algorithms are based on different calculations of the second moment matrices. One is based on stacking the columns of matrix observations, while another is by a more delicate projected approach. A well-known fact is that, in the presence of a change point, a factor model can be rewritten as a model with a larger number of common factors. In turn, this entails that, in the presence of a change point, the number of spiked eigenvalues in the second moment matrix of the data increases. Based on this, we propose two families of procedures - one based on the fluctuations of partial sums, and one based on extreme value theory - to monitor whether the first non-spiked eigenvalue diverges after a point in time in the monitoring horizon, thereby indicating the presence of a change point. This package also provides some simple functions for detecting and removing outliers, imputing missing entries and testing moments. See more details in He et al. (2021)<doi:10.48550/arXiv.2112.13479>.

License GPL-2 | GPL-3

Encoding UTF-8

Imports LaplacesDemon, RSpectra

Depends R (>= 3.5.0)

NeedsCompilation no

Repository CRAN

Date/Publication 2024-05-30 07:10:17 UTC

RoxygenNote 7.3.1

LazyData true

Contents

---

cv.table   *cv.table*

Description

A data matrix containing approximated critical values for the scaled Wiener process. A supplement to Table 1 in Horvath et al. (2004). The rows are for different gamma on a grid of 50 points in [0,0.49) with step size equal to 0.01, while the columns are for different alpha on a grid of 50 points in [0.001,0.5) with step size equal to 0.001. All the values are based on 100000 replications of simulated Wiener process on an equal grid of 20000 points in [0,1], thus can be slightly different from those in Horvath et al. (2004).

Usage

```
cv.table
```

Format

A matrix of size 50*500.

gamma the scaling parameter

alpha significance level (1-alpha)-th quantile

Details

#' @references Horvath L, Huskova M, Kokoszka P, et al (2004). Monitoring changes in linear models. *Journal of statistical Planning and Inference*, 126(1): 225-251.

---

gen.data	*generate data*

Description

This function generates matrix-valued time series under a two-way factor structure with/without a change point.

Usage

```
gen.data(
  Sample_T,
  p1,
  p2,
  k1,
  k2,
  tau = 0.5,
  change = 0,
  pp = 0.3,
  a = 0,
  cc = 0
)
```

Arguments

Sample_T	positive integer indicating the length of series.
p1	positive integer indicating the row dimension.
p2	positive integer indicating the column dimension.
k1	positive integer indicating the number of row factors.
k2	positive integer indicating the number of column factors.
tau	a real number in $(0, 1)$, indicating the location of change point, i.e., (τT).
change	the type of change, taking 0 for no change point, taking 1 for the case that the loading matrix **R** changes, taking other values for the case that a new row factor occurs.
pp	a number in $(0, 1]$, indicating the magnitude of the break. When $change = 1$, *pp* is the proportion of entries in **R** that changes; when *change* is not equal to 0 or 1, *pp* is the proportion of non-zero entries in the new factor loading.
a	a number in $[0, min(p_1, p_2))$, indicating the cross-sectional correlations of the idiosyncratic errors.
cc	a number in $[0, 1)$, indicating the AR(1) coefficient of the factor and error processes.

Details

See the paper He et al. (2021).

Value

a $T \times p1 \times p2$ array.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

References

He Y, Kong X, Trapani L, & Yu L(2021). Online change-point detection for matrix-valued time series with latent two-way factor structure. *arXiv preprint*, arXiv:2112.13479.

Examples

```
# set parameters
k1=3
k2=3
epsilon=0.05
Sample_T=50
p1=40
p2=20


# generate data
Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=1,pp=0.3)
print("the dimension of Y is:")
print(dim(Y))
```

---

gen.psi.tau.flat   *calculate eigenvalue series by "flat" method*

Description

This function calculates the rolling eigenvalue series for the monitoring process, based on the "flat" version of sample covanriance matrix.

Usage

```
gen.psi.tau.flat(
  Y,
  k,
  m = ceiling(max(20, (dim(Y)[3])^(r/(r + 2)))),
  delta,
  r = 8
)
```

Arguments

Y	the observed $T \times p1 \times p2$ array. T is the sample size, $p1$ and $p2$ are the row and column dimensions, respectively.
k	a positive integer determining which eigenvalue to monitor. $k = 1$ for the largest eigenvalue.
m	a positive integer (> 1) indicating the bandwidth of the rolling window.
delta	a number in $(0, 1)$ indicating the rescaling parameter for the eigenvalue. The default approach to calculate delta is in the paper He et al. (2021).
r	a positive integer indicating the order of the transformation function $g(x) = |x|^r$. Motivated by the paper, r should be chosen according to the moments of the data; see more details in He et al. (2021).

Details

The rolling eigenvalue series will start at the stage $m + 1$, with length $T - m$.

Value

a $(T - m) \times 3$ matrix, whose three columns are the original, rescaled, and transformed eigenvalue series, respectively.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

References

He Y, Kong X, Trapani L, & Yu L(2021). Online change-point detection for matrix-valued time series with latent two-way factor structure. *arXiv preprint*, arXiv:2112.13479.

Examples

```
## generate data
k1=3
k2=3
epsilon=0.05
Sample_T=50
p1=40
p2=20
kmax=8
r=8
m=p2

# generate data
Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=1,pp=0.3)

# calculate delta
temp=log(p1)/log(m*p2)
delta=epsilon*(temp<=0.5)+(epsilon+1-1/(2*temp))*(temp>0.5)
```

```
# calculate psi.tau
psi2=gen.psi.tau.flat(Y,k1+1,m,delta,r)
print(psi2)
```

---

gen.psi.tau.proj   *calculate eigenvalue series by projected method*

Description

This function calculates the rolling eigenvalue series for the monitoring process, based on the projected version of sample covanriance matrix.

Usage

```
gen.psi.tau.proj(
  Y,
  k,
  m = ceiling(max(20, (dim(Y)[3])^(r/(r + 2)))),
  delta,
  r = 8,
  kmax = 3
)
```

Arguments

Y	the observed $T \times p1 \times p2$ array. T is the sample size, $p1$ and $p2$ are the row and column dimensions, respectively.
k	a positive integer determining which eigenvalue to monitor. $k = 1$ for the largest eigenvalue.
m	a positive integer (> 1) indicating the bandwidth of the rolling windom.
delta	a number in $(0, 1)$ indicating the rescaling parameter for the eigenvalue. The default approach to calcualte delta is in the paper He et al. (2021).
r	a positive integer indicating the order of the transformation function $g(x) = |x|^r$. Motivated by the paper, r should be chosen according to the moments of the data; see more details in He et al. (2021).
kmax	a positive integer indicating the column number of the projection matrix, should be larger than 0 but smaller than $p2$.

Details

The rolling eigenvalue series will start at the stage $m + 1$, with length $T - m$.

Value

a $(T - m) \times 3$ matrix, whose three columns are the original, rescaled, and transformed eigenvalue series, respectively.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

References

He Y, Kong X, Trapani L, & Yu L(2021). Online change-point detection for matrix-valued time series with latent two-way factor structure. *arXiv preprint*, arXiv:2112.13479.

Examples

```
## generate data
k1=3
k2=3
epsilon=0.05
Sample_T=50
p1=40
p2=20
kmax=8
r=8
m=p2

# generate data
Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=1,pp=0.3)

# calculate delta
temp=log(p1)/log(m*p2)
delta=epsilon*(temp<=0.5)+(epsilon+1-1/(2*temp))*(temp>0.5)

# calculate psi.tau
psi2=gen.psi.tau.proj(Y,k1+1,m,delta,r,kmax)
print(psi2)
```

---

getcv   *calculate critical values*

Description

This function calculates critical values for the partial-sum and worst-case statistics.

Usage

```
getcv(alpha = 0.05, method = "ps", eta = 0.5, simul = 0)
```

Arguments

`alpha`	a number in $(0, 1)$, indicating the significance level of the test.
`method`	"ps" for the partial-sum staistic, others for the worst-case statistic.
`eta`	a number in $[0, 1]$, a scaling parameter required for "ps" method; see more details in He et al. (2021).
`simul`	logical value, woking only for "ps" method with η not equal to 0.5. When *simul* is true, the function will return approximated critical values based on 50000 replications of simulated Wiener process on a grid of 10000 points in $[0, 1]$. Otherwise, the function first checks for the nearest pair of (η, α) in the preserved *cv.table*, and then returns the corresponding critical value.

Details

For the partial-sum statistic with $\eta = 0.5$ or the worst-case statistic, the critical value is simply $-log(-log(1 - alpha))$. For the partial-sum statistic with η not equal to 0.5, the critical value of the scaled Wiener process is approximated by simulated data or from our preserved table *cv.table*, covering η in $[0.01, 0.49]$ with step size equal to 0.01 and α in $[0.001, 0.500]$ with step size equal to 0.001. See more details for the test statistics in He et al. (2021).

Value

a real number.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

References

He Y, Kong X, Trapani L, & Yu L(2021). Online change-point detection for matrix-valued time series with latent two-way factor structure. *arXiv preprint*, arXiv:2112.13479.

Examples

```
## Not run:
getcv(0.05,method="ps",eta=0.25)
getcv(0.05,method="ps",eta=0.25,simul=1)
getcv(0.10,method="wc")

## End(Not run)
```

---

impute.linear *impute missing entries by linear interpolation*

Description

This function imputes missing entries in a numeric vector by linear interpolation.

Usage

```
impute.linear(x)
```

Arguments

x   a numeric data vector, where NA indicates missing entries. The vector should contain at least two non-missing values.

Details

A missing entry will be imputed by linear interpolation with the two nearest values before and after it in the vector. When all the values before (after) it are missing, use the two nearest values after (before) it, instead.

Value

a new numeric vector, with the same size of the original vector, while all the missing entries have been imputed.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

Examples

```
a=c(NA,NA,3,NA,NA,6,NA,NA)
b=c(1,2,3,4.5,5,NA,6.5,7,NA)

impute.linear(a)
impute.linear(b)
```

| ITP_noproj | *testing the number of row factors- without projection* |

Description

This function tests whether the number of row factors is equal or larger than a given integer, under a two-way factor model, using flat version of sample covariance.

Usage

```
ITP_noproj(
  Y,
  k = 1,
  alpha = 0.05,
  epsilon = 0.05,
  r = 8,
  M = 100,
  S = 100,
  fq = 1/4
)
```

Arguments

Y	data, a $T \times p1 \times p2$ array.
k	an positive integer indicating which eigenvalue to test.
alpha	a number in (0,1), indicating the significance of the test.
epsilon	a small positive number in (0,1), indicating the size of scaling.
r	a positive number indicating the order of the power function for transforming the rescaled eigenvalue.
M	a large integer for the number of Gaussian variables in the randomized test.
S	another large integer for the number of replications in the strong rule. Usually $M = S = T$.
fq	a number in (0,0.5), controlling the threshold function of the strong rule.

Details

See He et al. (2023)

Value

a logical value. 1 for "the number of row factors is smaller than k". 0 for "at least k row factors exists".

References

He Y, Kong X, Trapani L, & Yu L (2023). One-way or two-way factor model for matrix sequences? *Journal of Econometrics*, 235(2), 1981-2004.

Examples

```
k1=3
k2=3
Sample_T=100
p1=40
p2=20

Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=0)
ITP_noproj(Y,k=1,M=Sample_T,S=Sample_T)
ITP_noproj(Y,k=4,M=Sample_T,S=Sample_T)
```

ITP_proj   *testing the number of row factors- with projection*

Description

This function tests whether the number of row factors is equal or larger than a given integer, under a two-way factor model, using projected version of sample covariance.

Usage

```
ITP_proj(
  Y,
  k = 1,
  alpha = 0.05,
  kmax = 4,
  epsilon = 0.05,
  r = 8,
  M = 100,
  S = 100,
  fq = 1/4
)
```

Arguments

Y	data, a $T \times p1 \times p2$ array.
k	an positive integer indicating which eigenvalue to test.
alpha	a number in (0,1), indicating the significance of the test.
kmax	a positive integer smaller than p2, indicating the upper bound for the factor numbers, and the dimension of projection matrix.
epsilon	a small positive number in (0,1), indicating the size of scaling.
r	a positive number indicating the order of the power function for transforming the rescaled eigenvalue.
M	a large integer for the number of Gaussian variables in the randomized test.

| S | another large integer for the number of replications in the strong rule. Usually $M = S = T$. |
| fq | a number in (0,0.5), controlling the threshold function of the strong rule. |

Details

See He et al. (2023)

Value

a logical value. 1 for "the number of row factors is smaller than k". 0 for "at least k row factors exists".

References

He Y, Kong X, Trapani L, & Yu L (2023). One-way or two-way factor model for matrix sequences? *Journal of Econometrics*, 235(2), 1981-2004.

Examples

```
k1=3
k2=3
Sample_T=100
p1=40
p2=20

Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=0)
ITP_proj(Y,k=1,M=Sample_T,S=Sample_T)
ITP_proj(Y,k=4,M=Sample_T,S=Sample_T)
```

---

kpe   *determine factor number - projected*

Description

This function determined the numbers of row and column factors for matrix variate data under a two-way factor model, using a projected method.

Usage

```
kpe(Y, kmax = 4, c = 0)
```

Arguments

Y	data, a $T \times p1 \times p2$ array.
kmax	a positive integer smaller than p2, indicating the upper bound for the factor numbers, and the dimension of projection matrix.
c	a non-negative but small number, to ensure the denominator of the eigenvalue ratio statistics is not 0.

Value

a 2-dimensional vector containing the estimated number of row and column factors, respectively.

References

Yu L, He Y, Kong X, & Zhang X (2022). Projected estimation for large-dimensional matrix factor models. *Journal of Econometrics*, 229(1),201-217.

Examples

```
k1=3
k2=3
Sample_T=100
p1=40
p2=20
kmax=8

Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=0)
kpe(Y,kmax)
```

KSTP   *determine row factor number - test*

Description

This function determines the number of row factors under a two-way factor structure, using randomized test method.

Usage

```
KSTP(
  Y,
  alpha = 0.05,
  type = "proj",
  kmax = 4,
  epsilon = 0.05,
  r = 8,
  M = 100,
  S = 100,
  fq = 1/4
)
```

Arguments

Y	data, a $T \times p1 \times p2$ array.
alpha	a number in (0,1), indicating the significance of the test.
type	indicates how to calculate the sample covariance. "flat" for the flat version, while others for the projected version.
kmax	a positive integer smaller than p2, indicating the upper bound for the factor numbers, and the dimension of projection matrix.
epsilon	a small positive number in (0,1), indicating the size of scaling.
r	a positive number indicating the order of the power function for transforming the rescaled eigenvalue.
M	a large integer for the number of Gaussian variables in the randomized test.
S	another large integer for the number of replications in the strong rule. Usually $M = S = T$.
fq	a number in (0,0.5), controlling the threshold function of the strong rule.

Details

See He et al. (2023)

Value

an integer for the number of row factors. To determine the number of column factors, just transpose the observation matrices.

References

He Y, Kong X, Trapani L, & Yu L (2023). One-way or two-way factor model for matrix sequences? *Journal of Econometrics*, 235(2), 1981-2004.

Examples

```
k1=3
k2=3
Sample_T=100
p1=40
p2=20

Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=0)
KSTP(Y)
KSTP(aperm(Y,c(1,3,2)))
```

---

moment.determine *determine the moment (largest) of the data samples*

Description

This function reports the largest moment that exists for a collection of data samples.

Usage

```
moment.determine(x, k.max = 8, alpha = 0.05, R = 400)
```

Arguments

x	a numeric vector of data samples.
k.max	a number indicating the upper bound, i.e., at most k.max-th moment exists.
alpha	a number in $(0, 1)$, indicating the significance level of the test.
R	the number of standard Gaussian variables generated in the randomized test; see also moment.test.

Details

The procedure will sequentially test the existence of the $4th, 6th, 8th, ...k.max-th$ moment, using the function $moment.test$ in the same package. As soon as the procedure finds that the $k-th$ moment does not exist, it stops and reports at most $(k-1)-th$ moment.

Value

an integer, indicating the largest moment that exists for the data samples.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

Examples

```
x=rt(10000,5)
moment.determine(x,10)

x=rt(10000,4)
moment.determine(x,10)
```

---

moment.test   *test whether the k-th moment exists*

Description

This function tests the existence of k-th moment by randomized method in Trapani (2016).

Usage

```
moment.test(x, k = 16, R = 400)
```

Arguments

x	a numeric vector of data samples.
k	a number no smaller than 4, indicating that the procedure will test the existence of the k-th moment when k is even. Otherwise, the procedure will test the existence of the k'-th moment, with $k' = round(k/2, 0) \times 2$.
R	the number of standard Gaussian variables generated in the randomized test; see more details in Trapani (2016).

Details

The procedure is adapted from Trapani (2016) with $\psi = 2$, where ψ is a tuning parameter to scale the sample moments defined in Section 3.1 of Trapani (2016). For simplicity, we only test the 4th, 6th, ... 2c-th moments.

Value

a scalar in $[0, 1]$, indicating the p-value of the test. The null hypothese is that the k-th moment doesn't exist. Therefore, a small p-value indicates the existense of the k-th moment.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

References

Trapani, L. (2016). Testing for (in) finite moments. *Journal of Econometrics*, 191(1), 57-68.

Examples

```
x=rt(10000,5)
moment.test(x,4)

x=rt(10000,4)
moment.test(x,4)
```

---

outlier.remove   *remove outliers*

Description

This function removes outliers in the data, which are far from the sample median.

Usage

```
outlier.remove(x, rg = 3)
```

Arguments

x   a numeric data vector.

rg   a positive number indicating how the outliers are defined; see details.

Details

An outlier is detected if it deviates from the sample median more than *rg* times interquantile range.

Value

a list containing the following:

x   a new data vector with outliers replaced with NA. The original missing values
                        in the data are preserved.

id   the locations of the outliers in the data vector.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

Examples

```
a=c(1:5,NA,10000)
outlier.remove(a,3)
```

---

test.multiple.robust   *robust test of multiple change point for matrix-valued online time series*

Description

This function tests multiple change points for matrix-valued online time series, under a two-way factor structure. A change point will be reported only when it's the majority vote in multiple replications. The function KSTP is used to determine the initial number of factors in each regime. This function only outputs the change points for row factors. For column factors, transpose the data.

Usage

```
test.multiple.robust(
  Y,
  k = 1,
  m = ceiling(max(20, (dim(Y)[3])^(r/(r + 2)))),
  epsilon1 = 0.25,
  epsilon2 = 0.05,
  r = 8,
  kmax = 4,
  type = "proj",
  method = "ps",
  eta = 0.25,
  cv = 2.386,
  S = 100,
  pr = 0.75
)
```

Arguments

Y	data, a $T \times p1 \times p2$ array.
k	a non-negative integer indicating the initial number of factors.
m	a positive integer (> 1) indicating the bandwidth of the rolling windom.
epsilon1	the rescaling parameter taking value in $(0, 1)$, for the test of new factors or the change of loading space; see He et al. (2021).
epsilon2	the rescaling parameter taking value in $(0, 1)$, for the test of vanishing factors; see He et al. (2021).
r	a positive integer indicating the order of the transformation function $g(x) = |x|^r$; see also gen.psi.tau.proj.
kmax	a positive number determining the column number of the projection matrix, should be larger than 0 but smaller than p_2, required when *type* not being "flat".
type	indicates how to calculate the sample covariance. "flat" for the flat version, while others for the projected version. See more details in He et al. (2021).

method	indicating the test statistic, "ps" for the partial-sum method; others for the worst-case method.
eta	a number between $[0, 1)$, indicating the parameter η used in the partial-sum statistic.
cv	critical value; see also `test.once.psi`.
S	an integer indicating the number of replications.
pr	an number in $(0, 1])$. The procedure reports a change point only when the proportion of positive votes is over *pr* in the *S* replications.

Details

#' See empirical study in He et al. (2021).

Value

a matrix with two columns. The first column reports the locations of change points. The second column reports the number of row factors after each change point.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

References

He Y, Kong X, Trapani L, & Yu L(2021). Online change-point detection for matrix-valued time series with latent two-way factor structure. *arXiv preprint*, arXiv:2112.13479.

Examples

```
## Not run:
k1=3
k2=3
Sample_T=100
p1=40
p2=40
kmax=8
r=8
m=p2

# generate data
Y1=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=1,pp=0.5)
Y2=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=0)
Y=array(rbind(matrix(Y1,Sample_T,p1*p2),matrix(Y2,Sample_T,p1*p2)),c(Sample_T*2,p1,p2))

# calculate cv for "ps" with eta=0.45 and "wc"
cv1=getcv(0.05,method="ps",eta=0.45)
cv2=getcv(0.05,method="wc")

# test with Y
test.multiple.robust(Y,k1,m,epsilon1=0.25,epsilon2=0.05,r,type="proj",kmax,method="ps")
```

```
test.multiple.robust(Y,k1,m,epsilon1=0.25,epsilon2=0.05,r,type="proj",kmax,method="wc",cv=cv2)

test.multiple.robust(Y,k1,m,epsilon1=0.25,epsilon2=0.05,r,type="flat",method="wc",cv=cv2)

test.multiple.robust(Y,k1,m,epsilon1=0.25,epsilon2=0.05,r,type="flat",method="ps",eta=0.45,cv=cv1)


## End(Not run)
```

---

test.once.flat	*test single change point for matrix-valued online time series -"flat" version*

Description

This function tests single change point for matrix-valued online time series, under a two-way factor structure, using "flat" sample covariance matrix.

Usage

```
test.once.flat(
  Y,
  k = 1,
  m = ceiling(max(20, (dim(Y)[3])^(r/(r + 2)))),
  epsilon = 0.05,
  r = 8,
  decrease = 0,
  method = "ps",
  eta = 0.25,
  cv = 2.386
)
```

Arguments

Y	data, a $T \times p1 \times p2$ array.
k	a positive integer indicating which eigenvalue to monitor. $k = 1$ for the largest eigenvalue.
m	a positive integer (> 1) indicating the bandwidth of the rolling windom.
epsilon	the rescaling parameter taking value in $(0, 1)$; see He et al. (2021).
r	a positive integer indicating the order of the transformation function $g(x) = |x|^r$; see also gen.psi.tau.proj.
decrease	a logical value. If *decrease=1*, testing the decrease of factor number.
method	indicating the test statistic, "ps" for the partial-sum method; others for the worst-case method.
eta	a number between $[0, 1)$, indicating the parameter η used in the partial-sum statistic.
cv	critical value; see also test.once.psi.

Details

See He et al. (2021).

Value

a list containing:

test	a logical value. 1 indicating the existence of change point, 0 indicating no change point.
loc	an integer larger than m, indicating the location of change point; or NA when no change point is reported.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

References

He Y, Kong X, Trapani L, & Yu L(2021). Online change-point detection for matrix-valued time series with latent two-way factor structure. *arXiv preprint*, arXiv:2112.13479.

Examples

```
k1=3
k2=3
epsilon=0.05
Sample_T=50
p1=40
p2=20
r=8
m=p2

# generate data
Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=1,pp=0.5)

# calculate cv for "ps" with eta=0.45 and "wc"
cv1=getcv(0.05,method="ps",eta=0.45)
cv2=getcv(0.05,method="wc")


## test with Y, flat version
test.once.flat(Y,k1+1,m,epsilon,r,0,method="ps",eta=0.25)


test.once.flat(Y,k1+1,m,epsilon,r,0,method="ps",eta=0.45,cv1)


test.once.flat(Y,k1+1,m,epsilon,r,0,method="wc",eta=0.5,cv2)
```

---

test.once.flat.robust   *robust test of single change point for matrix-valued online time series -"flat" version*

Description

Based on `test.once.flat`, this function repeats the randomized procedure multiple times and reports the majority vote, thus more robust.

Usage

```
test.once.flat.robust(
  Y,
  k = 1,
  m = ceiling(max(20, (dim(Y)[3])^(r/(r + 2)))),
  epsilon = 0.05,
  r = 8,
  decrease = 0,
  method = "ps",
  eta = 0.25,
  cv = 2.386,
  S = 100,
  pr = 0.75
)
```

Arguments

Y	data, a $T \times p1 \times p2$ array.
k	a positive integer indicating which eigenvalue to monitor. $k = 1$ for the largest eigenvalue.
m	a positive integer (> 1) indicating the bandwidth of the rolling windom.
epsilon	the rescaling parameter taking value in $(0, 1)$; see He et al. (2021).
r	a positive integer indicating the order of the transformation function $g(x) = |x|^r$; see also `gen.psi.tau.proj`.
decrease	a logical value. If *decrease=1*, testing the decrease of factor number.
method	indicating the test statistic, "ps" for the partial-sum method; others for the worst-case method.
eta	a number between $[0, 1)$, indicating the parameter η used in the partial-sum statistic.
cv	critical value; see also `test.once.psi`.
S	an integer indicating the number of replications.
pr	an number in $(0, 1]$. The procedure reports a change point only when the proportion of positive votes is over *pr* in the *S* replications.

Details

See He et al. (2021).

Value

a list containing:

test   a logical value. 1 indicating the existence of change point, 0 indicating no change point.

loc   an integer larger than m, indicating the median location of the change point among the positive votes in the *S* replications; or NA when no change point is reported.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

References

He Y, Kong X, Trapani L, & Yu L(2021). Online change-point detection for matrix-valued time series with latent two-way factor structure. *arXiv preprint*, arXiv:2112.13479.

Examples

```
## Not run:
k1=3
k2=3
epsilon=0.05
Sample_T=50
p1=40
p2=20
kmax=8
r=8
m=p2

# generate data
Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=1,pp=0.5)

# calculate cv for "ps" with eta=0.45 and "wc"
cv1=getcv(0.05,method="ps",eta=0.45)
cv2=getcv(0.05,method="wc")


## test with Y, flat version
test.once.flat.robust(Y,k1+1,m,epsilon,r,0,method="ps",eta=0.25)


test.once.flat.robust(Y,k1+1,m,epsilon,r,0,method="ps",eta=0.45,cv1)


test.once.flat.robust(Y,k1+1,m,epsilon,r,0,method="wc",eta=0.5,cv2)
```

```
## End(Not run)
```

| test.once.proj | *test single change point for matrix-valued online time series-projected version* |

Description

This function tests single change point for matrix-valued online time series, under a two-way factor structure, using projected sample covariance matrix.

Usage

```
test.once.proj(
  Y,
  k = 1,
  m = ceiling(max(20, (dim(Y)[3])^(r/(r + 2)))),
  epsilon = 0.05,
  r = 8,
  kmax = 3,
  decrease = 0,
  method = "ps",
  eta = 0.25,
  cv = 2.386
)
```

Arguments

Y	data, a $T \times p1 \times p2$ array.
k	a positive integer indicating which eigenvalue to monitor. $k = 1$ for the largest eigenvalue.
m	a positive integer (> 1) indicating the bandwidth of the rolling window.
epsilon	the rescaling parameter taking value in $(0, 1)$; see He et al. (2021).
r	a positive integer indicating the order of the transformation function $g(x) = |x|^r$; see also gen.psi.tau.proj.
kmax	a positive number determining the column number of the projection matrix, should be larger than 0 but smaller than $p2$.
decrease	a logical value. If *decrease=1*, testing the decrease of factor number.
method	indicating the test statistic, "ps" for the partial-sum method; others for the worst-case method.
eta	a number between $[0, 1)$, indicating the parameter η used in the partial-sum statistic.
cv	critical value; see also test.once.psi.

Details

See He et al. (2021).

Value

a list containing:

test	a logical value. 1 indicating the existence of change point, 0 indicating no change point.
loc	an integer larger than m, indicating the location of change point; or NA when no change point is reported.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

References

He Y, Kong X, Trapani L, & Yu L(2021). Online change-point detection for matrix-valued time series with latent two-way factor structure. *arXiv preprint*, arXiv:2112.13479.

Examples

```
k1=3
k2=3
epsilon=0.05
Sample_T=50
p1=40
p2=20
kmax=8
r=8

# generate data
Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=1,pp=0.5)

# calculate cv for "ps" with eta=0.45 and "wc"
cv1=getcv(0.05,method="ps",eta=0.45)
cv2=getcv(0.05,method="wc")


## test with Y, projection
test.once.proj(Y,k1+1,kmax=8)

m=p2
test.once.proj(Y,k1+1,m,epsilon,r,kmax,0,method="ps",eta=0.45,cv1)


test.once.proj(Y,k1+1,m,epsilon,r,kmax,0,method="wc",eta=0.5,cv2)
```

---

test.once.proj.robust   *robust test of single change point for matrix-valued online time series-projected version*

Description

Based on `test.once.proj`, this function repeats the randomized procedure multiple times and reports the majority vote, thus more robust.

Usage

```
test.once.proj.robust(
  Y,
  k = 1,
  m = ceiling(max(20, (dim(Y)[3])^(r/(r + 2)))),
  epsilon = 0.05,
  r = 8,
  kmax = 3,
  decrease = 0,
  method = "ps",
  eta = 0.25,
  cv = 2.386,
  S = 100,
  pr = 0.75
)
```

Arguments

Y	data, a $T \times p1 \times p2$ array.
k	a positive integer indicating which eigenvalue to monitor. $k = 1$ for the largest eigenvalue.
m	a positive integer (> 1) indicating the bandwidth of the rolling windom.
epsilon	the rescaling parameter taking value in $(0, 1)$; see He et al. (2021).
r	a positive integer indicating the order of the transformation function $g(x) = |x|^r$; see also `gen.psi.tau.proj`.
kmax	a positive number determining the column number of the projection matrix, should be larger than 0 but smaller than p_2.
decrease	a logical value. If *decrease=1*, testing the decrease of factor number.
method	indicating the test statistic, "ps" for the partial-sum method; others for the worst-case method.
eta	a number between $[0, 1)$, indicating the parameter η used in the partial-sum statistic.
cv	critical value; see also `test.once.psi`.
S	an integer indicating the number of replications.
pr	an number in $(0, 1]$). The procedure reports a change point only when the proportion of positive votes is over *pr* in the *S* replications.

Details

See He et al. (2021).

Value

a list containing:

test	a logical value. 1 indicating the existence of change point, 0 indicating no change point.
loc	an integer larger than m, indicating the median location of the change point among the positive votes in the S replications; or NA when no change point is reported.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

References

He Y, Kong X, Trapani L, & Yu L(2021). Online change-point detection for matrix-valued time series with latent two-way factor structure. *arXiv preprint*, arXiv:2112.13479.

Examples

```
## Not run:
k1=3
k2=3
epsilon=0.05
Sample_T=50
p1=40
p2=20
kmax=8
r=8
m=p2

# generate data
Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=1,pp=0.5)

# calculate cv for "ps" with eta=0.45 and "wc"
cv1=getcv(0.05,method="ps",eta=0.45)
cv2=getcv(0.05,method="wc")


## test with Y, projection
test.once.proj.robust(Y,k1+1,m,epsilon,r,kmax,0,method="ps",eta=0.25)


test.once.proj.robust(Y,k1+1,m,epsilon,r,kmax,0,method="ps",eta=0.45,cv1)


test.once.proj.robust(Y,k1+1,m,epsilon,r,kmax,0,method="wc",eta=0.5,cv2)
```

```
## End(Not run)
```

---

test.once.psi	*test single change point for matrix-valued online data given rolling eigenvalue series*

Description

This function tests single change point for matrix-valued online time series, under a two-way factor structure, given the transformed eigenvalue series.

Usage

```
test.once.psi(m = 20, psi, method = "ps", eta = 0.25, cv = 2.386)
```

Arguments

m	a positive integer (> 1) indicating the bandwidth of the rolling windom.
psi	the transformed eigenvalue series, produced by gen.psi.tau.flat or gen.psi.tau.proj, with length $T - m$.
method	indicating the test statistic, "ps" for the partial-sum method, while others for the worst-case method.
eta	a number between $[0, 1)$, indicating the parameter η used in the partial-sum statistic.
cv	critical value, related to the significance level and test statistic. The default cv is from Horvath et al. (2004), and only works for $\eta = 0.25$ or $\eta = 0.75$. For other cases, generate the critical value first by function [getcv](). Note that for the partial-sum statistic with η not equal to 0.5, the critical values are approximated by simulated data, thus can be slightly different from those in Horvath et al. (2004).

Details

See He et al. (2021).

Value

a list containing:

test	a logical value. 1 indicating the existence of change point, 0 indicating no change point.
loc	an integer larger than m, indicating the location of change point; or NA when no change point is reported.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

References

Horvath L, Huskova M, Kokoszka P, et al (2004). Monitoring changes in linear models. *Journal of statistical Planning and Inference*, 126(1): 225-251.

He Y, Kong X, Trapani L, & Yu L(2021). Online change-point detection for matrix-valued time series with latent two-way factor structure. *arXiv preprint*, arXiv:2112.13479.

Examples

```
k1=3
k2=3
epsilon=0.05
Sample_T=50
p1=40
p2=20
kmax=8
r=8
m=p2

# generate data
Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=1,pp=0.5)

# calculate delta
temp=log(p1)/log(m*p2)
delta=epsilon*(temp<=0.5)+(epsilon+1-1/(2*temp))*(temp>0.5)

# calculate psi.tau
psi1=gen.psi.tau.proj(Y,k1+1,m,delta,r,kmax)
psi2=gen.psi.tau.flat(Y,k1+1,m,delta,r)

# calculate cv for "ps" with eta=0.45 and "wc"
cv1=getcv(0.05,method="ps",eta=0.45)
cv2=getcv(0.05,method="wc")


# test with psi1
test.once.psi(m,psi1[,3],method="ps",eta=0.45,cv1)

test.once.psi(m,psi1[,3],method="wc",eta=0.5,cv2)


# test with psi2
test.once.psi(m,psi2[,3],method="ps",eta=0.45,cv1)

test.once.psi(m,psi2[,3],method="wc",eta=0.5,cv2)
```

---

test.once.psi.robust *robust test of single change point for matrix-valued online data given rolling eigenvalue series*

Description

Based on `test.once.psi`, this function repeats the randomized procedure multiple times and reports the majority vote, thus more robust.

Usage

```
test.once.psi.robust(
  m = 20,
  psi,
  method = "ps",
  eta = 0.25,
  cv = 2.386,
  S = 100,
  pr = 0.75
)
```

Arguments

m	a positive integer (> 1) indicating the bandwidth of the rolling windom.
psi	the transformed eigenvalue series, produced by gen.psi.tau.flat or gen.psi.tau.proj, with length $T - m$.
method	indicating the test statistic, "ps" for the partial-sum method, while others for the worst-case method.
eta	a number between $[0, 1)$, indicating the parameter η used in the partial-sum statistic.
cv	critical value; see also `test.once.psi`.
S	an integer indicating the number of replications.
pr	an number in $(0, 1]$. The procedure reports a change point only when the proportion of positive votes is over *pr* in the *S* replications.

Details

See He et al. (2021).

Value

a list containing:

test	a logical value. 1 indicating the existence of change point, 0 indicating no change point.

loc an integer larger than m, indicating the median location of the change point among the positive votes in the S replications; or NA when no change point is reported.

Author(s)

Yong He, Xinbing Kong, Lorenzo Trapani, Long Yu

References

He Y, Kong X, Trapani L, & Yu L(2021). Online change-point detection for matrix-valued time series with latent two-way factor structure. *arXiv preprint*, arXiv:2112.13479.

Examples

```
k1=3
k2=3
epsilon=0.05
Sample_T=50
p1=40
p2=20
kmax=8
r=8
m=p2

# generate data
Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=1,pp=0.5)

# calculate delta
temp=log(p1)/log(m*p2)
delta=epsilon*(temp<=0.5)+(epsilon+1-1/(2*temp))*(temp>0.5)

# calculate psi.tau
psi1=gen.psi.tau.proj(Y,k1+1,m,delta,r,kmax)
psi2=gen.psi.tau.flat(Y,k1+1,m,delta,r)

# calculate cv for "ps" with eta=0.45 and "wc"
cv1=getcv(0.05,method="ps",eta=0.45)
cv2=getcv(0.05,method="wc")


# test with psi1
test.once.psi.robust(m,psi1[,3],method="ps",eta=0.45,cv1,S=100,pr=0.75)

test.once.psi.robust(m,psi1[,3],method="wc",eta=0.5,cv2,S=100,pr=0.75)


# test with psi2
test.once.psi.robust(m,psi2[,3],method="ps",eta=0.45,cv1,S=100,pr=0.75)

test.once.psi.robust(m,psi2[,3],method="wc",eta=0.5,cv2,S=100,pr=0.75)
```

`var.exp`   *explanatory power of factors*

Description

This function calculates the cumulative explanatory power of the leading row factors, in terms of the explained variance, under a two-way factor structure.

Usage

```
var.exp(Y, k = 2, type = "proj", kmax = 4, plot = 0)
```

Arguments

Y	data, a $T \times p1 \times p2$ array.
k	a positive integer indicating the number of factors investigated, should be smaller than p1.
type	indicates how to calculate the sample covariance. "flat" for the flat version, while others for the projected version.
kmax	a positive integer smaller than p2, indicating the upper bound for the factor numbers, and the dimension of projection matrix.
plot	a logical value. When *plot=1*, a figure of the cumulative explanatory power will be plotted, with x axis being the number of factors, and y axis being the cumulative explained variance.

Value

a vector with k entries, corresponding to the cumulative explanatory power of the leading k factors.

Examples

```
k1=3
k2=3
Sample_T=100
p1=40
p2=20

Y=gen.data(Sample_T,p1,p2,k1,k2,tau=0.5,change=0)
var.exp(Y,k=5,plot=1)
```

Index