Package ‘InspectChangepoint’

January 9, 2020

Type Package

Title High-Dimensional Changepoint Estimation via Sparse Projection

Version 1.1

Date 2020-01-09

Author Tengyao Wang and Richard Samworth

Maintainer Tengyao Wang <tengyao.wang@ucl.ac.uk>

Imports stats, graphics, MASS

Description Provides a data-driven projection-based method for estimating changepoints in high-dimensional time series. Multiple changepoints are estimated using a (wild) binary segmentation scheme.

License GPL-3

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-01-09 13:00:05 UTC

R topics documented:

`compute.threshold` *Computing threshold used in* `inspect`

Description

The threshold level to be used in `inspect` is computed via Monte Carlo simulation of multivariate time series that do not contain any changepoints.

Usage

```
compute.threshold(n, p, nrep = 100)
```

Arguments

`n`	Time length of the observation.
`p`	Dimension of the multivariate time series.
`nrep`	Number of Monte Carlo repetition to be used.

Value

A numeric value indicating the threshold level that should be used based on the Monte Carlo simulation.

Examples

```
compute.threshold(n=200, p=50)
```

`cusum.transform` *CUSUM transformation*

Description

Performing CUSUM transformation to the input matrix of multivariate time series. If the input is a vector, it is treated as a matrix of one row.

Usage

```
cusum.transform(x)
```

Arguments

x	input matrix

Details

For any integers p and n, the CUSUM transformation $T_{p,n} : R^{p\times n} \to R^{p\times(n-1)}$ is defined by

$$[T_{p,n}(M)]_{j,t} := \sqrt{t(n-t)/n}\left(\frac{1}{n-t}\sum_{r=t+1}^{n} M_{j,r} - \frac{1}{t}\sum_{r=1}^{t} M_{j,r}\right).$$

Value

The transformed matrix is returned. Note that the returned matrix has the same number of rows but one fewer columns compared with the input matrix.

Examples

```
x <- matrix(rnorm(20),4,5)
cusum.transform(x)
```

---

inspect *Informative sparse projection for estimation of changepoints (inspect)*

Description

This is the main function of the package InspectChangepoint. The function `inspect` estimates the locations of multiple changepoints in the mean structure of a multivariate time series. Multiple changepoints are estimated using a (wild) binary segmentation scheme, whereas each segmentation step uses the `locate.change` function.

Usage

```
inspect(x, lambda, threshold, schatten=c(1,2), M)
```

Arguments

x	The input data matrix of a high-dimensional time series, with each component time series stored as a row.
lambda	Regularisation parameter used in `locate.change`. If no value is supplied, the dafault value is chosen to be log(log(n)*p/2), where p and n are the number of rows and columns of the data matrix x respectively.
threshold	Threshold level for testing whether an identified changepoint is a true changepoint. If no value is supplied, the threshold level is computed via Monte Carlo simulation of 100 repetitions from the null model.
schatten	The Schatten norm constraint to use in the `locate.change` function. Default is schatten = 2, i.e. a Frobenius norm constraint.
M	The Monte Carlo parameter used for wild binary segmentation. Default is M = 0, which means a classical binary segmentation scheme is used.

Details

The input time series is first standardised using the `rescale.variance` function. Recursive calls of the `locate.change` function then segments the multivariate time series using (wild) binary segmentation. A changepoint at time z is defined here to mean that the time series has constant mean structure for time up to and including z and constant mean structure for time from z+1 onwards.

More details about model assumption and theoretical guarantees can be found in Wang and Samworth (2016). Note that Monte Carlo computation of the threshold value can be slow, especially for large p. If `inspect` is to be used multiple times with the same (or similar) data matrix size, it is better to precompute the threshold level via Monte Carlo simulation by calling the `compute.threshold` function.

Value

The return value is an S3 object of class 'inspect'. It contains a list of two objeccts:

- x The input data matrix
- changepoints A matrix with three columns. The first column contains the locations of estimated changepoints sorted in increasing order; the second column contains the maximum CUSUM statistics of the projected univariate time series associated with each estimated changepoint; the third column contains the depth of binary segmentation for each detected changepoint.

References

Wang, T. and Samworth, R. J. (2018) High dimensional changepoint estimation via sparse projection. *J. Roy. Statist. Soc., Ser. B*, **80**, 57–83.

Examples

```
n <- 500; p <- 100; ks <- 30; zs <- c(125,250,375)
varthetas <- c(0.1,0.15,0.2); overlap <- 0.5
obj <- multi.change(n, p, ks, zs, varthetas, overlap)
x <- obj$x
threshold <- compute.threshold(n,p)
ret <- inspect(x, threshold = threshold)
ret
summary(ret)
plot(ret)
```

`locate.change` *Single changepoint estimation*

Description

Estimate the location of one changepoint in a multivariate time series. It uses the function `sparse.svd` to estimate the best projection direction, then using univariate CUSUM statistics of the projected time series to estimate the changepoint location.

Usage

```
locate.change(x, lambda, schatten = 2, sample.splitting = FALSE,
  standardize.series = FALSE, view.cusum = FALSE)
```

Arguments

`x`	A (p x n) data matrix of multivariate time series, each column represents a data point
`lambda`	Regularisation parameter. If no value is supplied, the dafault value is chosen to be sqrt(log(log(n)*p/2)) for p and n number of rows and columns of the data matrix x respectively.
`schatten`	The Schatten norm constraint to use in the `sparse.svd` function. Default is schatten = 2, i.e. a Frobenius norm constraint.
`sample.splitting`	Whether the changepoint should be estimated via sample splitting. The theoretical result is proven only for the sample splitted version of the algorithm. However, the default setting in practice is without sample splitting.
`standardize.series`	Whether the given time series should be standardised before estimating the projection direction. Default is FALSE, i.e. the input series is assume to have variance 1 in each coordinate.
`view.cusum`	Whether to show a plot of the projected CUSUM series

Value

A list of two items:

- changepoint - A single integer value estimate of the changepoint location is returned. If the estimated changepoint is z, it means that the multivariate time series is piecewise constant up to z and from z+1 onwards.
- cusum - The maximum absolute CUSUM statistic of the projected univariate time series associated with the estimated changepoint.
- vector.proj - the vector of projection, which is proportional to an estimate of the vector of change.

References

Wang, T., Samworth, R. J. (2016) High-dimensional changepoint estimation via sparse projection. Arxiv preprint: arxiv1606.06246.

Examples

```
n <- 2000; p <- 1000; k <- 32; z <- 400; vartheta <- 0.12; sigma <- 1; shape <- 3
noise <- 0; corr <- 0
obj <- single.change(n,p,k,z,vartheta,sigma,shape,noise,corr)
x <- obj$x
locate.change(x)
```

`multi.change` *Generating a high-dimensional time series with multiple changepoints*

Description

The data matrix is generated via X = mu + W, where mu is the mean structure matrix that captures the changepoint locations and sparsity structure, and W is a random noise matrix having independent N(0,sigma^2) entries.

Usage

```
multi.change(n, p, ks, zs, varthetas, sigma = 1, overlap = 0,
  shape = 3)
```

Arguments

`n`	Time length of the observation
`p`	Dimension of the multivariate time series
`ks`	A vector describing the number of coordinates that undergo a change in each changepoint. If only a scalar is supplied, each changepoint will have the same number of coordinates that undergo a change.
`zs`	A vector describing the locations of the changepoints.
`varthetas`	A vector describing the root mean squared change magnitude in coordinates that undergo a change for each changepoint. If only a scalar is supplied, each changepoint will have the same signal strength value.
`sigma`	noise level
`overlap`	A number between 0 and 1. The proportion of overlap in the signal coordinates for successive changepoints.
`shape`	How the signal strength is distributed across signal coordinates. When shape = 0, all signal coordinates are changed by the same amount; when shape = 1, their signal strength are proportional to 1, sqrt(2), ..., sqrt(k); when shape = 2, they are proportional to 1, 2, ..., k; when shape = 3, they are proportional to 1, 1/sqrt(2), ..., 1/sqrt(k).

Value

An S3 object of the class 'hdchangeseq' is returned.

- x - The generated data matrix
- mu - The mean structure of the data matrix

See Also

`plot.hdchangeseq`

Examples

```
n <- 2000; p <- 200; ks <- 40;
zs <- c(500,1000,1500); varthetas <- c(0.1,0.15,0.2); overlap <- 0.5
obj <- multi.change(n, p, ks, zs, varthetas, overlap)
plot(obj, noise = TRUE)
```

`PiS` *Matrix projection onto the nuclear norm unit sphere*

Description

Projection (with respect to the inner product defined by the Frobenius norm) of a matrix onto the unit sphere defined by the nuclear norm.

Usage

```
PiS(M)
```

Arguments

`M`	Input matrix

Details

This is an auxiliary function used by the `InspectChangepoint` package. The projection is achieved by first performing a singular value decomposition, then projecting the vector of singular values onto the standard simplex, and finally using singular value decomposition in reverse to build the projected matrix.

Value

A matrix of the same dimension as the input is returned.

Examples

```
M <- matrix(rnorm(20),4,5)
PiS(M)
```

PiW — *Projection onto the standard simplex*

Description

The input vector is projected onto the standard simplex, i.e. the set of vectors of the same length as the input vector with non-negative entries that sum to 1.

Usage

```
PiW(v)
```

Arguments

v	Input vector

Details

This is an auxiliary function used by the `InspectChangepoint` package.

Value

A vector in the standard simplex that is closest to the input vector is returned.

References

Chen, Y. and Ye, X. (2011) Projection onto a simplex. arXiv preprint, arxiv:1101.6081.

Examples

```
v <- rnorm(10)
PiW(v)
```

plot.hdchangeseq — *Plot function for 'hdchangeseq' class*

Description

Visualising the high-dimensional time series in an 'hdchangeseq' class object. The data matrix or its mean structure is visualised using a grid of coloured rectangles with colours corresponding to the value contained in corresponding coordinates. A heat-spectrum (red to white for values low to high) is used to convert values to colours.

Usage

```
## S3 method for class 'hdchangeseq'
plot(x, noise = TRUE, shuffle = FALSE, ...)
```

Arguments

`x`	An object of 'hdchangeseq' class
`noise`	If noise == TRUE, the data matrix is plotted, otherwise, only the mean structure is plotted.
`shuffle`	Whether to shuffle the rows of the plotted matrix.
`...`	Other graphical parameters are not used.

Examples

```
n <- 2000; p <- 200; ks <- 40; zs <- c(500,1000,1500)
varthetas <- c(0.1,0.15,0.2); overlap <- 0.5
obj <- multi.change(n, p, ks, zs, varthetas, overlap)
plot(obj, noise = TRUE)
```

`plot.inspect` *Plot function for 'inspect' class objects*

Description

Plot function for 'inspect' class objects

Usage

```
## S3 method for class 'inspect'
plot(x, ...)
```

Arguments

`x`	an 'inspect' class object
`...`	other arguments to be passed to methods are not used

See Also

`inspect`

power.method — *Power method for finding the leading eigenvector of a symmetric matrix*

Description

Power method for finding the leading eigenvector of a symmetric matrix

Usage

```
power.method(A, eps = 1e-10, maxiter = 10000)
```

Arguments

`A`	a square symmetric matrix
`eps`	tolerance for convergence (in Frobenius norm)
`maxiter`	maximum iteration

Value

a unit-length leading eigenvector of A

print.inspect — *Print function for 'inspect' class objects*

Description

Print function for 'inspect' class objects

Usage

```
## S3 method for class 'inspect'
print(x, ...)
```

Arguments

`x`	an 'inspect' class object
`...`	other arguments to be passed to methods are not used

See Also

`inspect`

`rescale.variance` *Noise standardisation for multivariate time series.*

Description

Each row of the input matrix is normalised by the estimated standard deviation computed through the median absolute deviation of increments.

Usage

```
rescale.variance(x)
```

Arguments

x	An input matrix of real values.

Details

This is an auxiliary function used by the `InspectChangepoint` package.

Value

A rescaled matrix of the same size is returned.

Examples

```
x <- matrix(rnorm(40),5,8) * (1:5)
x.rescaled <- rescale.variance(x)
x.rescaled
```

`single.change` *Generating high-dimensional time series with exactly one change in the mean structure*

Description

The data matrix is generated via X = mu + W, where mu is the mean structure matrix that captures the changepoint location and sparsity structure, and W is a random noise matrix.

Usage

```
single.change(n, p, k, z, vartheta, sigma = 1, shape = 3, noise = 0,
  corr = 0)
```

Arguments

`n`	Time length of the observation
`p`	Dimension of the multivariate time series
`k`	Number of coordinates that undergo a change
`z`	Changepoint location, a number between 1 and n-1.
`vartheta`	The root mean squared change magnitude in coordinates that undergo a change
`sigma`	noise level, see `noise` for more details.
`shape`	How the signal strength is distributed across signal coordinates. When shape = 0, all signal coordinates are changed by the same amount; when shape = 1, their signal strength are proportional to 1, sqrt(2), ..., sqrt(k); when shape = 2, they are proportional to 1, 2, ..., k; when shape = 3, they are proportional to 1, 1/sqrt(2), ..., 1/sqrt(k).
`noise`	Noise structure of the multivarite time series. For noise = 0, 0.5, 1, columns of W have independent multivariate normal distribution with covariance matrix Sigma. When noise = 0, Sigma = sigma^2 * I_p; when noise = 0.5, noise has local dependence structure given by Sigma_i,j = sigma*corr^|i-j|; when noise = 1, noise has global dependence structure given by matrix(corr,p,p)+diag(p)*(1-corr))) * sigma. When noise = 2, rows of the W are independent and each having an AR(1) structure given by W_j,t = W_j,t-1 * sqrt(corr) + rnorm(sd = sigma) * sqrt(1-corr). For noise = 3, 4, entries of W have i.i.d. uniform distribution and exponential distribution respectively, each centred and rescaled to have zero mean and variance sigma^2.
`corr`	Used to specify correlation structure in the noise. See `noise` for more details.

Value

An S3 object of the class 'hdchangeseq' is returned.

- x - The generated data matrix
- mu - The mean structure of the data matrix

See Also

`plot.hdchangeseq`

Examples

```
n <- 2000; p <- 100; k <- 10; z <- 800; vartheta <- 1; sigma <- 1
shape <- 3; noise <- 0; corr <- 0
obj <- single.change(n,p,k,z,vartheta,sigma, shape, noise, corr)
plot(obj, noise = TRUE)
```

sparse.svd *Computing the sparse leading left singular vector of a matrix*

Description

Estimating the sparse left leading singular vector by first computing a maximiser Mhat of the convex problem

$$< Z, M > -\lambda |M|_1$$

subject to the Schatten norm constraint |M|_schatten <= 1 using alternating direction method of multipliers (ADMM). Then the leading left singular vector of Mhat is returned.

Usage

```
sparse.svd(Z, lambda, schatten = c(1, 2), tolerance = 1e-05,
  max.iter = 10000)
```

Arguments

Z	Input matrix whose left leading singular vector is to be estimated.
lambda	Regularisation parameter
schatten	Schatten norm constraint to be used. Default uses Schatten-2-norm, i.e. the Frobenius norm. Also possible to use Schatten-1-norm, the nuclear norm.
tolerance	Tolerance criterion for convergence of the ADMM algorithm. Not used when shatten=2.
max.iter	Maximum number of iteration in the ADMM algorithm. Not used when shatten=2.

Details

In case of schatten = 2, a closed-form solution for Mhat using matrix soft thresholding is possible. We use the closed-form solution instead of the ADMM algorithm to speed up the computation.

Value

A vector that has the same length as nrow(Z) is returned.

Examples

```
Z <- matrix(rnorm(20),4,5)
lambda <- 0.5
sparse.svd(Z, lambda)
```

summary.inspect — *Summary function for 'inspect' class objects*

Description

Summary function for 'inspect' class objects

Usage

```
## S3 method for class 'inspect'
summary(object, ...)
```

Arguments

`object`	an 'inspect' class object
`...`	other arguments to be passed to methods are not used

See Also

`inspect`

vector.norm — *Norm of a vector*

Description

Calculate the entrywise L_q norm of a vector or a matrix

Usage

```
vector.norm(v, q = 2, na.rm = FALSE)
```

Arguments

`v`	a vector of real numbers
`q`	a nonnegative real number or Inf
`na.rm`	boolean, whether to remove NA before calculation

Value

the entrywise L_q norm of a vector or a matrix `vector.soft.thresh` *Soft thresholding a vector*

Description

entries of v are moved towards 0 by the amount lambda until they hit 0.

Usage

```
vector.soft.thresh(x, lambda)
```

Arguments

`x`	a vector of real numbers
`lambda`	soft thresholding value

Value

a vector of the same length

Index