Package 'changepoint.geo'

August 31, 2025

Type Package

Title Geometrically Inspired Multivariate Changepoint Detection

Version 1.0.3

Date 2025-08-12

Maintainer Rebecca Killick <r.killick@lancaster.ac.uk>

URL https://github.com/grundy95/changepoint.geo/

Description Implements the high-dimensional changepoint detection method GeomCP and the related mappings used for changepoint detection. These methods view the changepoint problem from a geometrical viewpoint and aim to extract relevant geometrical features in order to detect changepoints. The geomcp() function should be your first point of call. References: Grundy et al. (2020) <doi:10.1007/s11222-020-09940-y>.

Depends R(>= 3.6), changepoint, changepoint.np, methods, ggplot2

Imports Rdpack

RdMacros Rdpack

Suggests testthat, MASS

License GPL

LazyLoad yes

NeedsCompilation no

Author Thomas Grundy [aut],
Rebecca Killick [cre, ths]

Repository CRAN

Date/Publication 2025-08-31 20:50:02 UTC

Contents

changepoint.geo-package

Geometrically Inspired Multivariate Changepoint Detection

Description

Implements the high-dimensional changepoint detection method GeomCP (Grundy et al. 2020) and the related mappings used for changepoint detection. These methods view the changepoint problem from a geometrical viewpoint and aim to extract relevant geometrical features in order to detect changepoints. The geomcp() function should be your first point of call.

Details

Package:	changepoint.geo
Type:	Package
Title:	Geometrically Inspired Multivariate Changepoint Detection
Version:	1.0.3
Date:	2025-08-12
Authors@R:	c(person("Thomas","Grundy",role=c("aut")), person("Rebecca","Killick",role=c("cre","ths"),email="r.k
Maintainer:	Rebecca Killick <r.killick@lancaster.ac.uk>
URL:	https://github.com/grundy95/changepoint.geo/
Description:	Implements the high-dimensional changepoint detection method GeomCP and the related mappings use
Depends:	R(>= 3.6), changepoint, changepoint.np, methods, ggplot2
Imports:	Rdpack
RdMacros:	Rdpack
Suggests:	testthat, MASS
License:	GPL
LazyLoad:	yes
NeedsCompilation:	no
Author:	Thomas Grundy [aut], Rebecca Killick [cre, ths]
Repository:	CRAN

Author(s)

Thomas Grundy [aut], Rebecca Killick [cre, ths]

Maintainer: Rebecca Killick <r.killick@lancaster.ac.uk>

References

Grundy T, Killick R, Mihalyov G (2020). "High-dimensional changepoint detection via a geometrically inspired mapping." *Stat Comput*, **0**(0). doi:10.1007/s1122202009940y.

Killick R, Fearnhead P, Eckley IA (2012). "Optimal detection of changepoints with a linear computational cost." *J. Am. Stat. Assoc.*, **107**(500), 1590–1598.

See Also

geomcp

Examples

```
X <- rbind(matrix(rnorm(100*50),ncol=50),matrix(rnorm(100*50,0,2),ncol=50))
ans <- geomcp(X)
summary(ans)
```

---

angle.mapping *Principle angle mapping*

Description

Calculates the principle angle of each time vector to the reference vector. Note data points are not translated in this function and error checking is not performed!

Usage

```
angle.mapping(X,ref.vec.value=rep(1,length(X[1,])))
```

Arguments

X	A matrix containing the centralized data of size n by p.
ref.vec.value	A vector containing the reference vector from which the principle angle will be calculated.

Details

This function calculates the principle angle between each time vector and the reference vector. Note the translation within geomcp happens outside of the function.

Value

A vector of length n is returned with the angle mapping for each time point.

Author(s)

Thomas Grundy

See Also

geomcp

Examples

```
X <- rbind(matrix(rnorm(100*50,2),ncol=50),matrix(rnorm(100*50,2,4),ncol=50))
ref.vec.value <- rep(1,50)
ans <- angle.mapping(X,ref.vec.value)
ans
```

`class_input` *Input all required arguments into cpt.geo class - Only intended for developer use.*

Description

This function helps to input all necessary information into the correct format for the `cpt.geo` class.

This function is called when `class=TRUE`. This is not intended for use by regular users of the package. It is exported for developers to call directly for speed and convenience.

WARNING: No checks on arguments are performed!

Usage

```
class_input(data.set, distance, angle, penalty, pen.value, test.stat,
    msl, nquantiles, dist.cpts, ang.cpts)
```

Arguments

`data.set`	Data set used in multivariate changepoint analysis, see `geomcp` for further details.
`distance`	Vector containing the distance mappings
`angle`	Vector containing the angle mappings
`penalty`	Penalty used as a text string, see `geomcp` for further details.
`pen.value`	Numerical penalty value used in the univariate analysis of the mapped time series.
`test.stat`	The assumed test statistic/distribution of the univariate mapped time series as a text string. See `geomcp` for further details.
`msl`	Minimum segment length used in the analysis (positive integer).
`nquantiles`	Number of quantiles used in empirical distribution if using `test.stat='Empirical'`.
`dist.cpts`	Changepoint locations in distance mappings.
`ang.cpts`	Changepoint locations in angle mappings.

Details

This function takes all the input required for the `cpt.geo` class and enters it into an object.

This function is exported for developer use only. It does not perform any checks on inputs and is simply a convenience function for converting the output of the worker functions into a nice format for the `cpt.geo` class.

Value

An object of class `cpt.geo` filled with the given attributes.

Author(s)

Thomas Grundy

See Also

geomcp,plot-methods,cpt.geo

Examples

```
#This function should only be used by developers, see its use in geomcp.
```

---

distance.mapping *Euclidean distance mapping*

Description

Calculates the Euclidean distance of each time vector to the reference vector. Note data points are not translated in this function and error checking is not performed!

Usage

```
distance.mapping(X,ref.vec.value=rep(1,length(X[1,])))
```

Arguments

X	A matrix containing the centralized data of size n by p.
ref.vec.value	A vector containing the reference vector from which the euclidean distance will be calculated.

Details

This function calculates the Euclidean distance between each time vector and the reference vector. Note the translation within geomcp happens outside of the function.

Value

A vector of length n is returned with the distance mapping for each time point.

Author(s)

Thomas Grundy

See Also

geomcp

Examples

```
X <- rbind(matrix(rnorm(100*50,2),ncol=50),matrix(rnorm(100*50,2.5),ncol=50))
ref.vec.value <- rep(1,50)
ans <- distance.mapping(X,ref.vec.value)
ans
```

---

geomcp *Multivariate changepoint detection via two geometric mappings*

Description

Implements the GeomCP algorithm (Grundy et al. 2020). This algorithm performs two geometric mappings on multivariate time series based on the Euclidean distance and principle angle between each time vector and a pre-specified reference vector. The univariate changepoint detection method PELT, (Killick et al. 2012), is then performed on the two mappings to identify changepoints which correspond to those in the original multivariate time series.

Usage

```
geomcp(X, penalty = "MBIC", pen.value = 0, test.stat = "Normal", msl = 2,
       nquantiles = 1,MAD=FALSE,ref.vec='Default',ref.vec.value=0)
```

Arguments

X	A matrix containing the data which is of size n by p. Each row is a new time point and each column is a different series.
penalty	Penalty choice for univariate changepoint analysis of mappings. Choice of "MBIC", "SIC", "BIC", "Hannan-Quinn","Manual". If "Manual" is specified, the manual penalty is contained in the pen.value parameter.
pen.value	The value of the penalty when using the "Manual" penalty option - this can be a numeric value or text giving the formula to use, see cpt.meanvar for more details.
test.stat	The assumed test statistic/distribution of the mapped data. Currently only "Normal" and "Empirical" are supported.
msl	Positive integer giving the minimum segment length (no. of observations between changes), default is 1.
nquantiles	Only required for test.stat="Empirical". Number of quantiles used in estimating the empirical likelihood.
MAD	Logical. If TRUE transforms each series by subtracting the median of the series and dividing by the median absolute deviation.
ref.vec	Choice of "Default" or "Manual". If "Default" is specified the vector of ones is used as the reference vector. If "Manual" is selected, the manual reference vector is contained in the ref.vec.value parameter.
ref.vec.value	The vector that is used as the reference vector for calculating distances and angles from.

Details

This function centralizes all time vectors using the given reference vector and then performs the distance and angle mappings from the reference vector. The univariate changepoint method PELT is then used to detect changepoints in the distance and angle mappings which correspond to changes in the multivariate time series.

Value

An object of S4 class "cpt.geo" is returned. The slots `dist.cpts` and `ang.cpts` return the changepoints identified in each measure.

Author(s)

Thomas Grundy

References

Grundy T, Killick R, Mihalyov G (2020). "High-dimensional changepoint detection via a geometrically inspired mapping." *Stat Comput*, **0**(0). doi:10.1007/s1122202009940y.

Killick R, Fearnhead P, Eckley IA (2012). "Optimal detection of changepoints with a linear computational cost." *J. Am. Stat. Assoc.*, **107**(500), 1590–1598.

See Also

`cpt.geo`,`plot-methods`

Examples

```
##Variance change in all series
set.seed(1)
X <- rbind(matrix(rnorm(100*50),ncol=50),matrix(rnorm(100*50,0,2),ncol=50))
ans <- geomcp(X)
summary(ans)
plot(ans)

##Mean change in 50% of series with a manual reference vector and non-parametric univariate
##changepoint detection using 10 quantiles with a BIC penalty and min seg length of 5
set.seed(1)
Y <- rbind(matrix(rnorm(100*20),ncol=20),cbind(matrix(rnorm(100*10),ncol=10),
       matrix(rnorm(100*10,1),ncol=10)))
res <- geomcp(Y,penalty='Manual',pen.value=30,test.stat='Empirical',nquantiles=10,ref.vec='Manual',
      ref.vec.value=seq(1,10,length.out=20),msl=5)
summary(res)
plot(res)

##Different plot types for above example
#Plots mappings and changepoints
plot(res,plot.type='mappings')
#Heatmap of data with changepoints not shown
plot(res,plot.type='full.data',changepoints=FALSE,scale.series=TRUE)
#Specific series with mappings and changepoints shown.
```

```
plot(res,plot.type='series',show.series=c(1,5,10),add.mappings=TRUE)
```

Index