

Package ‘multiSeg’

November 25, 2017

Type Package

Title Multidimensional changepoint detection with dynamic programming

Version 0.1

Date 2017-11-21

Author Vincent Runge

Maintainer Vincent Runge<runge.vincent@gmail.com>

Description Experimental package to test different pruning methods in the multidimensional change-point problem. The core function of the package are implemented in C++.

License GPL-2

Imports Rcpp (>= 0.12.3), graphics, utils, stats

LinkingTo Rcpp

RoxygenNote 6.0.1

Archs x64

R topics documented:

| | |
|------------------------|---|
| changepoints | 1 |
| dataG | 2 |
| multiSeg | 3 |
| pruning | 3 |
| segmentation | 4 |
| Index | 5 |

| | |
|--------------|-------------------------------|
| changepoints | <i>Vector of changepoints</i> |
|--------------|-------------------------------|

Description

Vector of changepoints

Usage

changepoints(res)

Arguments

res A multiSeg object (obtained with function multiSeg)

Value

The vector of changepoints

Examples

```
n <- 500
var <- 1
means <- matrix(c(0,1,2,1,0,1,2,2,0,0,1,1),3,4)
changes <- c(0.3,0.6,1)
dataG(n,means,changes, var)
file = paste(getwd(),"/dataG.txt",sep = "")
res <- multiSeg(file,4*2*log(n),1)
changepoints(res)
```

dataG

Multidimensional Gaussian Data Generator

Description

Simulating Multidimensional Gaussian Data with a model of segmentation, that is a given matrix of means and a vector of changepoints.

Usage

```
dataG(n, means, chgpt, sigma = 1)
```

Arguments

n number of observations to generate.

means matrix of means, the number of columns = number of variables = number of dimensions. The i-th row corresponds to the i-th segment mean in all dimensions.

chgpt vector of changepoints (an increasing sequence of integers).

sigma standard deviation of the gaussian model

Value

The matrix of simulated data + the matrix is saved in the file dataG.txt at the current path (obtained by getwd())

Examples

```
data <- dataG(500, matrix(c(0,1,2,1,0,1,2,2,0,0,1,1),3,4),c(0.3,0.6,1))
head(data)
```

| | |
|----------|---|
| multiSeg | <i>Multidimensional multiple changepoint algorithm with pruning</i> |
|----------|---|

Description

Dynamic programming algorithm in multiple dimensions with pruning

Usage

```
multiSeg(file, beta, type = 3, show = 0)
```

Arguments

| | |
|------|--|
| file | the complete path + file name of the data to use |
| beta | the penalty coefficient |
| type | an integer equal to 0,1,2 or 3. Complexity of the pruning. (see function pruning). If type = i, we use pruning method from 1 to i |
| show | an integer. Following the actual iteration during computation. Display every 'show' iteration. |

Value

A multiSeg object = (chgpt, means, nb, type, delay). With n the number of observations, all these elements are vector of size n except means which is a matrix of size nxp. 'Chgpt' is a vector of last changepoint at each position. 'mean' is the matrix of means on each last segment at each position. 'nb' is a vector whose elements are the number of present lastchangepoint candidates at each position. 'type' is a vector saving how the candidate was pruned 'delay' is the number of iterations before pruning for each data point.

Examples

```
n <- 100
var <- 1
means <- matrix(c(0,1,2,1,0,1,2,2,0,0,1,1),3,4)
changes <- c(0.3,0.6,1)
data <- dataG(n,means,changes, var)
file = paste(getwd(),"/dataG.txt",sep = "")
multiSeg(file,4*2*log(n),1)
```

| | |
|---------|----------------------------------|
| pruning | <i>Plots of the pruning path</i> |
|---------|----------------------------------|

Description

Displaying the properties of the pruning

Usage

```
pruning(res)
```

Arguments

res A multiSeg object (obtained with function multiSeg)

Value

a plot with 3 graphics. First plot = number of candidates (nb) Second plot (in red) = delay before pruning for each candidate (delay) Third plot = cumulative sums for all type of pruning (0, 1, 2 or 3). type = 0 (black curve): no pruning type = 1 (red curve): radius < 0 type = 2 (green curve): intersection = emptyset type = 3 (blue curve): candidate hidden by one sphere of the past

Examples

```
n <- 500
var <- 1
means <- matrix(c(0,1,2,1,0,1,2,2,0,0,1,1),3,4)
changes <- c(0.3,0.6,1)
data <- dataG(n,means,changes, var)
file = paste(getwd(),"/dataG.txt",sep = "")
res <- multiSeg(file,4*2*log(n),1)
pruning(res)
res$type
```

segmentation

Screening of the segmentation

Description

Display the segments and changepoints

Usage

```
segmentation(file, res)
```

Arguments

file the path + name of the data file
res A multiSeg object (obtained with function multiSeg)

Examples

```
n <- 500
var <- 1
means <- matrix(c(0,1,2,1,0,1,2,2,0,0,1,1),3,4)
changes <- c(0.3,0.6,1)
dataG(n,means,changes, var)
file = paste(getwd(),"/dataG.txt",sep = "")
res <- multiSeg(file,4*2*log(n),1)
segmentation(file,res)
```

Index

changepoints, [1](#)

dataG, [2](#)

multiSeg, [3](#)

pruning, [3](#)

segmentation, [4](#)