Package ‘ecp’

September 27, 2016

Type Package

Title Non-Parametric Multiple Change-Point Analysis of Multivariate Data

Version 3.0.0

Date 2016-09-26

Author Nicholas A. James, Wenyu Zhang and David S. Matteson

Maintainer Nicholas A. James <nj89@cornell.edu>

Description Implements various procedures for finding multiple change-points. Two methods make use of dynamic programming and probabilistic pruning, with no distributional assumptions other than the existence of certain absolute moments in one method. Hierarchical and exact search methods are included. All methods return the set of estimated change-points as well as other summary information.

License GPL (>= 2)

Depends R (>= 3.00), Rcpp

Suggests mvtnorm, MASS, combinat, R.rsp

Imports stats, utils

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

VignetteBuilder R.rsp

Date/Publication 2016-09-27 08:41:46

R topics documented:

ACGH .......................................................... 2
DJIA .......................................................... 3
e.agglomerative ........................................ 3
e.cp3o ......................................................... 5
e.divisive ................................................... 7
ks.cp3o ....................................................... 9
Description

Micro-array data for 43 different individuals with a bladder tumor.

Usage

data(ACGH)

Format

A list with the following components.

data: The micro-array data for 43 individuals. This information is stored in a 2215 by 43 matrix.
individual: A numeric vector indicating which individuals’ micro-array data are present.

Source

Bleakley K., Vert J.-P. (2011), The group fused Lasso for multiple change-point detection


References

Bleakley K., Vert J.-P. (2011), The group fused Lasso for multiple change-point detection


Examples

data(ACGH, package="ecp")
DJIA

Dow Jones Industrial Average Index

Description

The weekly log returns for the Dow Jones Industrial Average index from April 1990 to January 2012.

Usage

data(DJIA)

Format

A list with the following components.

dates: A character vector of dates associated with each observation in the returns series.
index: Weekly log returns from April 1990 to January 2012 of the DOW 30 index.
market: Weekly log returns from April 1990 to January 2012, for the companies in the DOW 30 apart from Kraft.

Source

http://research.stlouisfed.org/fred2/series/DJIA/downloaddata

References


Examples

data(DJIA, package="ecp")

e.agglo

ENERGY AGGLOMERATIVE

Description

An agglomerative hierarchical estimation algorithm for multiple change point analysis.

Usage

e.agglo(X, member=1:nrow(X), alpha=1, penalty=function(cps){0})
Arguments

\textbf{x} \hspace{1cm} A \ T \times \ d \ matrix \ containing \ the \ length \ T \ time \ series \ with \ d-dimensional \ observations.

\textbf{member} \hspace{1cm} Initial \ membership \ vector \ for \ the \ time \ series.

\textbf{alpha} \hspace{1cm} Moment \ index \ used \ for \ determining \ the \ distance \ between \ and \ within \ clusters.

\textbf{penalty} \hspace{1cm} Function \ used \ to \ penalize \ the \ obtained \ goodness-of-fit \ statistics. \ This \ function \ takes \ as \ its \ input \ a \ vector \ of \ change \ point \ locations (cps).

Details

Homogeneous clusters are created based on the initial clustering provided by the \textit{member} argument. In each iteration, clusters are merged so as to maximize a goodness-of-fit statistic. The computational complexity of this method is $O(T^2)$, where $T$ is the number of observations.

Value

Returns a list with the following components.

\textbf{merged} \hspace{1cm} A (T-1) x 2 matrix indicating which segments were merged at each step of the agglomerative procedure.

\textbf{fit} \hspace{1cm} Vector showing the progression of the penalized goodness-of-fit statistic.

\textbf{progression} \hspace{1cm} A T x (T+1) matrix showing the progression of the set of change points.

\textbf{cluster} \hspace{1cm} The estimated cluster membership vector.

\textbf{estimates} \hspace{1cm} The location of the estimated change points.

Author(s)

Nicholas A. James

References


See Also

e.divisive
Examples

```r
set.seed(100)
mem = rep(c(1,2,3,4),times=c(10,10,10,10))
x = as.matrix(c(rnorm(10,0,1),rnorm(20,2,1),rnorm(10,-1,1)))
y = e.agglo(X=x,member=mem,alpha=1,penalty=function(cp,Xts) 0)
y$estimates
```

## Not run:
# Multivariate spatio-temporal example
# You will need the following packages:
# mvtnorm, combinat, and MASS
library(mvtnorm); library(combinat); library(MASS)
set.seed(2013)
lambda = 1500 # overall arrival rate per unit time
muA = c(-7,-7); muB = c(0,0); muC = c(5.5,0)
covA = 25*diag(2); covB = matrix(c(9,0,0,1),2); covC = matrix(c(9,.9,.9,2)
time.interval = matrix(c(0,1,3,4.5,1,3,4.5,7),2)
# mixing coefficients
mixing.coef = rbind(c(1/3,1/3,1/3),c(.2,.5,.3),c(.35,.3,.35),c(.2,.3,.5))
stppData = NULL
for(i in 1:4){
  count = rpois(1, lambda*diff(time.interval[i]))
  Z = rmult2n(n = count, p = mixing.coef[i,])
  S = rbind(rmvnorm(Z[1],muA,covA), rmvnorm(Z[2],muB,covB),rmvnorm(Z[3],muC,covC))
  X = cbind(rep(i,count), runif(n = count, time.interval[1,1], time.interval[1,2]), S)
  stppData = rbind(stppData, X[order(X[,2]),])
}
member = as.numeric(cut(stppData[,2], breaks = seq(0,7,by=1/12)))
output = e.agglo(X=stppData[,3:4],member=member,alpha=1,penalty=function(cp,Xts) 0)
## End(Not run)
```

---

**e.cp3o**

*CHANGE POINTS ESTIMATION BY PROBABILISTICALLY PRUNED OBJECTIVE (VIA E-STATISTIC)*

### Description

An algorithm for multiple change point analysis that uses dynamic programming and probabilistic pruning. The E-statistic is used as the goodness-of-fit measure.

### Usage

```r
e.cp3o(Z, K=1, delta=29, alpha=1, eps=0.01, verbose=FALSE)
```
**Arguments**

- **Z**: A T x d matrix containing the length T time series with d-dimensional observations.
- **K**: The maximum number of change points.
- **delta**: The window size used to calculate the approximate test statistic. This also corresponds to one less than the minimum segment size.
- **alpha**: The moment index used for determining the distance between and within segments.
- **eps**: The epsilon probability used for the probabilistic pruning procedure.
- **verbose**: A flag indicating if status updates should be printed.

**Details**

Segmentations are found through the use of dynamic programming and probabilistic pruning. The computational complexity of this method is \(O(KT^2)\), where \(K\) is the maximum number of change points, and \(T\) is the number of observations.

**Value**

The returned value is a list with the following components.

- **number**: The estimated number of change points.
- **estimates**: The location of the change points estimated by the procedure.
- **gofM**: A vector of goodness of fit values for differing number of change points. The first entry corresponds to when there is only a single change point, the second for when there are two, and so on.
- **cpLoc**: A list of all the optimal change point locations for differing numbers of change points. The first component corresponds to when there is only one change point, the second for when there are two change points, and so on.
- **time**: The total amount of time take to estimate the change point locations.

**Author(s)**

Nicholas A. James

**References**


Examples

```r
set.seed(400)
x1 = matrix(c(rnorm(100), rnorm(100, 3), rnorm(100, 0, 2)))
y1 = e.cp3o(Z=x1, K=7, delta=29, alpha=1, eps=0.01, verbose=FALSE)
# View estimated change point locations
y1$estimates
x2 = rbind(MASS::mvrnorm(100, c(0, 0), diag(2)), MASS::mvrnorm(100, c(2, 2), diag(2)))
y2 = e.cp3o(Z=x2, K=4, delta=29, alpha=1, eps=0.01, verbose=FALSE)
# View estimated change point locations
y2$estimates
# View all possible segmentations for differing numbers of change points
y2$cpLoc
```

---

e.divisive

**ENERGY DIVISIVE**

Description

A divisive hierarchical estimation algorithm for multiple change point analysis.

Usage

```r
e.divisive(X, sig.lvl=.05, R=199, k=NULL, min.size=30, alpha=1)
```

Arguments

- **X** A T x d matrix containing the length T time series with d-dimensional observations.
- **sig.lvl** The level at which to sequentially test if a proposed change point is statistically significant.
- **R** The maximum number of random permutations to use in each iteration of the permutation test. The permutation test p-value is calculated using the method outlined in Gandy (2009).
- **k** Number of change point locations to estimate, suppressing the permutation based testing. If k=NULL then only the statistically significant estimated change points are returned.
- **min.size** Minimum number of observations between change points.
- **alpha** The moment index used for determining the distance between and within segments.

Details

Segments are found through the use of a binary bisection method and a permutation test. The computational complexity of this method is $O(kT^2)$, where $k$ is the number of estimated change points, and $T$ is the number of observations.
Value
The returned value is a list with the following components.

- `k.hat`: The number of clusters within the data created by the change points.
- `order.found`: The order in which the change points were estimated.
- `estimates`: Locations of the statistically significant change points.
- `considered.last`: Location of the last change point, that was not found to be statistically significant at the given significance level.
- `permutations`: The number of permutations performed by each of the sequential permutation test.
- `cluster`: The estimated cluster membership vector.
- `p.values`: Approximate p-values estimated from each permutation test.

Author(s)
Nicholas A. James

References


See Also
e agglo

Examples
```r
set.seed(100)
x1 = matrix(c(rnorm(100), rnorm(100, 3), rnorm(100, 0, 2)))
y1 = e.divisive(X=x1, sig.lvl=0.05, R=199, k=NULL, min.size=30, alpha=1)
x2 = rbind(MASS::mvrnorm(100, c(0, 0), diag(2)), MASS::mvrnorm(100, c(2, 2), diag(2)))
y2 = e.divisive(X=x2, sig.lvl=0.05, R=499, k=NULL, min.size=30, alpha=1)
```
CHANGE POINTS ESTIMATION BY PROBABILISTICALLY PRUNED OBJECTIVE (VIA KOLMOGOROV-SMIRNOV STATISTIC)

Description

An algorithm for multiple change point analysis that uses dynamic programming and probabilistic pruning. The Kolmogorov-Smirnov statistic is used as the goodness-of-fit measure.

Usage

ks.cp3o(Z, K=1, minsize=30, eps=0.01, verbose=False)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>A T x d matrix containing the length T time series with d-dimensional observations.</td>
</tr>
<tr>
<td>K</td>
<td>The maximum number of change points.</td>
</tr>
<tr>
<td>minsize</td>
<td>The minimum segment size.</td>
</tr>
<tr>
<td>eps</td>
<td>The epsilon probability used for the probabilistic pruning procedure.</td>
</tr>
<tr>
<td>verbose</td>
<td>A flag indicating if status updates should be printed.</td>
</tr>
</tbody>
</table>

Details

Segmentations are found through the use of dynamic programming and probabilistic pruning.

Value

The returned value is a list with the following components.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>The estimated number of change points.</td>
</tr>
<tr>
<td>estimates</td>
<td>The location of the change points estimated by the procedure.</td>
</tr>
<tr>
<td>gofM</td>
<td>A vector of goodness of fit values for differing number of change points. The first entry corresponds to when there is only a single change point, the second for when there are two, and so on.</td>
</tr>
<tr>
<td>cpLoc</td>
<td>A list of all the optimal change point locations for differing numbers of change points. The first component corresponds to when there is only one change point, the second for when there are two change points, and so on.</td>
</tr>
<tr>
<td>time</td>
<td>The total amount to time take to estimate the change point locations.</td>
</tr>
</tbody>
</table>

Author(s)

Wenyu Zhang
References

Examples

```r
set.seed(400)
x = matrix(c(rnorm(100), rnorm(100, 3), rnorm(100, 0, 2)))
y = ks.cp3o(Z=x, K=7, minsize=30, eps=0.01, verbose=FALSE)
# View estimated change point locations
y$estimates
# View all possible segmentations for differing numbers of change points
y$cpLoc
```
Index

*Topic agglomerative
e.agglo, 3
*Topic datasets
   ACGH, 2
   DJIA, 3
*Topic divisive
e.divisive, 7
*Topic dynamic
e.cp3o, 5
   ks.cp3o, 9
*Topic hierarchical
e.agglo, 3
   e.divisive, 7
*Topic probabilistic
e.cp3o, 5
   ks.cp3o, 9
*Topic pruning
e.cp3o, 5
   ks.cp3o, 9

ACGH, 2
DJIA, 3
e.agglo, 3, 8
e.cp3o, 5
e.divisive, 4, 7
ks.cp3o, 9