Package ‘ecp’

November 11, 2019

Type Package

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

Version 3.1.2

Date 2019-11-10

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

Maintainer Wenyu Zhang <wz258@cornell.edu>

Description Implements various procedures for finding multiple change-points. Two methods make use of dynamic programming and 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

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

VignetteBuilder R.rsp

Date/Publication 2019-11-11 06:20:14 UTC

R topics documented:

`ACGH` *Bladder Tumor Micro-Array Data*

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' mico-array data are present.

Source

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

N. Stransky, C. Vallot, F. Reyal, I. Bernard-Pierrot, S.G. Diez de Mediana, R. Segraves, Y. de Rycke, P. Elvin, A. Cassidy, C. Sparaggon, A. Graham, j. Southgate, B. Asselain, Y. Allory, C. C. Addou, D. G. Albertson, J.-P. Thiery, D. K. Chopin, D. Pinkel, and F. Radvanyi. Regional copy number-independent deregulation of transcription in cancer. Nat. Genet., 38(12):1386-1396, Dec 2006

References

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

Nicholas A. James, David S. Matteson (2014). "ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data.", "Journal of Statistical Software, 62(7), 1-25", URL "http://www.jstatsoft.org/v62/i07/"

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

Nicholas A. James, David S. Matteson (2014). "ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data.", "Journal of Statistical Software, 62(7), 1-25", URL "http://www.jstatsoft.org/v62/i07/"

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

`X`	A T x d matrix containing the length T time series with d-dimensional observations.
`member`	Initial membership vector for the time series.
`alpha`	Moment index used for determining the distance between and within clusters.
`penalty`	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 *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.

`merged`	A (T-1) x 2 matrix indicating which segments were merged at each step of the agglomerative procedure.
`fit`	Vector showing the progression of the penalized goodness-of-fit statistic.
`progression`	A T x (T+1) matrix showing the progression of the set of change points.
`cluster`	The estimated cluster membership vector.
`estimates`	The location of the estimated change points.

Author(s)

Nicholas A. James

References

Matteson D.S., James N.A. (2013). A Nonparametric Approach for Multiple Change Point Analysis of Multivariate Data.

Nicholas A. James, David S. Matteson (2014). "ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data.", "Journal of Statistical Software, 62(7), 1-25", URL "http://www.jstatsoft.org/v62/i07/"

See Also

`e.divisive`

Rizzo M.L., Szekely G.L. (2005). Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method. Journal of Classification. pp. 151 - 183.

Rizzo M.L., Szekely G.L. (2010). Disco analysis: A nonparametric extension of analysis of variance. The Annals of Applied Statistics. pp. 1034 - 1055.

Examples

```
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,.9,9),2)
time.interval = matrix(c(0,1,3,4.5,1,3,4.5,7),4,2)
#mixing coefficents
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 = rmultz2(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[i,1],
time.interval[i,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 PRUNED OBJECTIVE (VIA E-STATISTIC)*

Description

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

Usage

```
e.cp3o(Z, K=1, minsize=30, alpha=1, 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.
minsize	The minimum segment size.
alpha	The moment index used for determining the distance between and within segments.
verbose	A flag indicating if status updates should be printed.

Details

Segmentations are found through the use of dynamic programming and pruning. For long time series, consider using e.cp3o_delta.

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	The list of locations of change points estimated by the procedure for different numbers of change points up to K.
time	The total amount to time take to estimate the change point locations.

Author(s)

Nicholas A. James, Wenyu Zhang

References

W. Zhang, N. A. James and D. S. Matteson, "Pruning and Nonparametric Multiple Change Point Detection," 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, 2017, pp. 288-295.

See Also

Rizzo M.L., Szekely G.L (2005). Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method. Journal of Classification.

Rizzo M.L., Szekely G.L. (2010). Disco analysis: A nonparametric extension of analysis of variance. The Annals of Applied Statistics.

Examples

```
set.seed(400)
x1 = matrix(c(rnorm(50),rnorm(50,3)))
y1 = e.cp3o(Z=x1, K=2, minsize=30, alpha=1, verbose=FALSE)
#View estimated change point locations
y1$estimates
```

`e.cp3o_delta` *CHANGE POINTS ESTIMATION BY PRUNED OBJECTIVE (VIA E-STATISTIC)*

Description

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

Usage

```
e.cp3o_delta(Z, K=1, delta=29, alpha=1, 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 calculate the complete portion of our 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.
`verbose`	A flag indicating if status updates should be printed.

Details

Segmentations are found through the use of dynamic programming and pruning. Between-segment distances are calculated only using points within a window of the segmentation point. 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`	The list of locations of change points estimated by the procedure for different numbers of change points up to K.
`time`	The total amount to time take to estimate the change point locations.

Author(s)

Nicholas A. James, Wenyu Zhang

References

W. Zhang, N. A. James and D. S. Matteson, "Pruning and Nonparametric Multiple Change Point Detection," 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, 2017, pp. 288-295.

See Also

Rizzo M.L., Szekely G.L (2005). Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method. Journal of Classification.

Rizzo M.L., Szekely G.L. (2010). Disco analysis: A nonparametric extension of analysis of variance. The Annals of Applied Statistics.

Examples

```
set.seed(400)
x1 = matrix(c(rnorm(100),rnorm(100,3),rnorm(100,0,2)))
y1 = e.cp3o_delta(Z=x1, K=7, delta=29, alpha=1, verbose=FALSE)
#View estimated change point locations
y1$estimates
```

`e.divisive` *ENERGY DIVISIVE*

Description

A divisive hierarchical estimation algorithm for multiple change point analysis.

Usage

```
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 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

Matteson D.S., James N.A. (2013). A Nonparametric Approach for Multiple Change Point Analysis of Multivariate Data.

Nicholas A. James, David S. Matteson (2014). "ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data.", "Journal of Statistical Software, 62(7), 1-25", URL "http://www.jstatsoft.org/v62/i07/"

See Also

e.agglo

Gandy, A. (2009) "Sequential implementation of Monte Carlo tests with uniformly bounded resampling risk." Journal of the American Statistical Association.

Rizzo M.L., Szekely G.L (2005). Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method. Journal of Classification.

Rizzo M.L., Szekely G.L. (2010). Disco analysis: A nonparametric extension of analysis of variance. The Annals of Applied Statistics.

Examples

```
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)
```

---

kcpa — *Kernel Change Point Analysis*

Description

An algorithm for multiple change point analysis that uses the 'kernel trick' and dynamic programming.

Usage

```
kcpa(X, L, C)
```

Arguments

X	A T x d matrix containing the length T time series with d-dimensional observations.
L	The maximum number of change points.
C	The constant used to penalize the inclusion of additional change points in the fitted model.

Details

Segments are found through the use of dynamic programming and the kernel trick.

Value

If the algorithm determines that the best fit is obtained through using k change points then the returned value is an array of length k, containing the change point locations.

Author(s)

Nicholas A. James

References

Harchaoui Z., Moulines E., Francis R.B (2009). Kernel Change-point Analysis. Advances in Neural Information Processing Systems.

`ks.cp3o` *CHANGE POINTS ESTIMATION BY PRUNED OBJECTIVE (VIA KOLMOGOROV-SMIRNOV STATISTIC)*

Description

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

Usage

```
ks.cp3o(Z, K=1, minsize=30, 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.
`minsize`	The minimum segment size.
`verbose`	A flag indicating if status updates should be printed.

Details

Segmentations are found through the use of dynamic programming and pruning. For long time series, consider using ks.cp3o_delta.

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`	The list of locations of change points estimated by the procedure for different numbers of change points up to K.
`time`	The total amount to time take to estimate the change point locations.

Author(s)

Wenyu Zhang

References

W. Zhang, N. A. James and D. S. Matteson, "Pruning and Nonparametric Multiple Change Point Detection," 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, 2017, pp. 288-295.

See Also

Kifer D., Ben-David S., Gehrke J. (2004). Detecting change in data streams. International Conference on Very Large Data Bases.

Examples

```
set.seed(400)
x = matrix(c(rnorm(100),rnorm(100,3),rnorm(100,0,2)))
y = ks.cp3o(Z=x, K=7, minsize=30, verbose=FALSE)
#View estimated change point locations
y$estimates
```

---

ks.cp3o_delta — *CHANGE POINTS ESTIMATION BY PRUNED OBJECTIVE (VIA KOLMOGOROV-SMIRNOV STATISTIC)*

Description

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

Usage

```
ks.cp3o_delta(Z, K=1, minsize=30, 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.
`minsize`	The minimum segment size. This is also the window size used to calculate between-segment distances.
`verbose`	A flag indicating if status updates should be printed.

Details

Segmentations are found through the use of dynamic programming and pruning. Between-segment distances are calculated only using points within a window of the segmentation point.

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`	The list of locations of change points estimated by the procedure for different numbers of change points up to K.
`time`	The total amount to time take to estimate the change point locations.

Author(s)

Wenyu Zhang

References

W. Zhang, N. A. James and D. S. Matteson, "Pruning and Nonparametric Multiple Change Point Detection," 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, 2017, pp. 288-295.

See Also

Kifer D., Ben-David S., Gehrke J. (2004). Detecting change in data streams. International Conference on Very Large Data Bases.

Examples

```
set.seed(400)
x = matrix(c(rnorm(100),rnorm(100,3),rnorm(100,0,2)))
y = ks.cp3o_delta(Z=x, K=7, minsize=30, verbose=FALSE)
#View estimated change point locations
y$estimates
```

Index