Package ‘breakpoint’

January 13, 2016

Type Package
Title An R Package for Multiple Break-Point Detection via the Cross-Entropy Method
Version 1.2
Date 2016-01-11
Author Priyadarshana W.J.R.M. and Georgy Sofronov
Maintainer Priyadarshana W.J.R.M. <mjayawardana@swin.edu.au>
Description Implements the Cross-Entropy (CE) method, which is a model based stochastic optimization technique to estimate both the number and their corresponding locations of break-points in continuous and discrete measurements (Priyadarshana and Sofronov (2015), Priyadarshana and Sofronov (2012a), Priyadarshana and Sofronov (2012b)).
License GPL (>= 2)
Depends R (>= 2.5)
Imports ggplot2 (>= 1.0.0), MASS, msm (>= 1.0.1), foreach (>= 1.2.0), parallel, doParallel (>= 1.0.10)
URL https://github.com/madawaweer/breakpoint
NeedsCompilation no
Repository CRAN
Date/Publication 2016-01-13 14:39:45

R topics documented:

breakpoint-package .......................................................... 2
CE.NB ............................................................................... 3
CE.NB.Init ................................................................. 5
CE.Normal.Init.Mean ..................................................... 8
CE.Normal.Init.MeanVar ............................................... 10
CE.Normal.Mean ......................................................... 13
CE.Normal.MeanVar .................................................... 15
CE.ZINB ........................................................................ 17
CE.ZINB.Init ........................................................... 20
ch1.GM03563 ............................................................ 23
profilePlot ............................................................. 24
Description

The breakpoint package implements variants of the Cross-Entropy (CE) method proposed in Priyadarshana and Sofronov (2015, 2012a and 2012b) to estimate both the number and the corresponding locations of break-points in biological sequences of continuous and discrete measurements. The proposed method primarily built to detect multiple break-points in genomic sequences. However, it can be easily extended and applied to other problems.

Details

- Package: breakpoint
- Type: Package
- Version: 1.2
- Date: 2016-01-11
- License: GPL 2.0

"breakpoint" package provides estimates on both the number as well as the corresponding locations of break-points. The algorithms utilize the Cross-Entropy (CE) method, which is a model-based stochastic optimization procedure to obtain the estimates on locations. Model selection procedures are used to obtain the number of break-points. Current implementation of the methodology works as an exact search method in estimating the number of break-points. However, it supports calculations if the initial locations are provided. A parallel implementation of the procedures can be carried-out in Unix/Linux/MAC OSX and WINDOWS OS with the use of "parallel" and "doParallel" packages.

Author(s)

Priyadarshana, W.J.R.M. and Sofronov, G.
Maintainer: Priyadarshana, W.J.R.M. <mjayawardana@swin.edu.au>

References


Multiple Break-point Detection via the CE Method with Negative Binomial Distribution

Description

Performs calculations to estimate both the number of break-points and their corresponding locations of discrete measurements with the CE method. Negative binomial distribution is used to model the over-dispersed discrete (count) data. This function supports for the simulation of break-point locations in the CE algorithm based on the four parameter beta distribution or truncated normal distribution. User can select either BIC or AIC to select the optimal number of break-points.

Usage

CE.NB(data, Nmax = 10, eps = 0.01, rho = 0.05, M = 200, h = 5, a = 0.8, b = 0.8, distyp = 1, penalty = "BIC", parallel = FALSE)

Arguments

data: data to be analysed. A single column array or a dataframe.
Nmax: maximum number of break-points. Default value is 10.
eps: the cut-off value for the stopping criterion in the CE method. Default value is 0.01.
rho: the fraction which is used to obtain the best performing set of sample solutions (i.e., elite sample). Default value is 0.05.
M: sample size to be used in simulating the locations of break-points. Default value is 200.
h: minimum aberration width. Default is 5.
a: a smoothing parameter value. It is used in the four parameter beta distribution to smooth both shape parameters. When simulating from the truncated normal distribution, this value is used to smooth the estimates of the mean values. Default is 0.8.
b: a smoothing parameter value. It is used in the truncated normal distribution to smooth the estimates of the standard deviation. Default is 0.8.
distyp: distribution to simulate break-point locations. Options: 1 = four parameter beta distribution, 2 = truncated normal distribution. Default is 1.
penalty User can select either BIC or AIC to obtain the number of break-points. Options: "BIC", "AIC". Default is "BIC".

parallel A logical argument specifying if parallel computation should be carried-out (TRUE) or not (FALSE). By default it is set as ‘FALSE’. In WINDOWS OS systems "snow" functionalities are used, whereas in Unix/Linux/MAC OSX "multicore" functionalities are used to carryout parallel computations with the maximum number of cores available.

Details

The negative binomial (NB) distribution is used to model the discrete (count) data. NB model is preferred over the Poission model when over-dispersion is observed in the count data. A performance function score (BIC or AIC) is calculated for each of the solutions generated by the statistical distribution (four parameter beta distribution or truncated normal distribution), which is used to simulate break-points from no break-point to the user provided maximum number of break-points (default is 10). The solution that minimizes the BIC/AIC with respect to the number of break-points is reported as the optimal solution. Finally, a list containing a vector of break-point locations, number of break-points, BIC/AIC values and log-likelihood value is returned in the console.

Value

A list is returned with following items:

- No.BPs The number of break-points in the data that is estimated by the CE method
- BP.Loc A vector of break-point locations
- BIC/AIC BIC/AIC value
- ll Loglikelihood of the optimal solution

Author(s)

Priyadarshana, W.J.R.M. <mjayawardana@swin.edu.au>

References


See Also

CE.NB.Init for CE with Negative binomial with initial locations,
CE.ZINB for CE with zero-inflated negative binomial,
CE.ZINB.Init for CE with zero-inflated negative binomial with initial locations,
profilePlot to obtain mean profile plot.

Examples

```r
### Simulated data example ###
segs <- 6 # Number of segments
M <- c(1500, 2200, 800, 2500, 1000, 2000) # Segment width
#true.locations <- c(1501, 3701, 4501, 7001, 8001) # True break-point locations
seg <- NULL
p <- c(0.45, 0.25, 0.4, 0.2, 0.3, 0.6) # Specification of p's for each segment
for(j in 1:segs){
  seg <- c(seg, rnbinom(M[j], size =10, prob = p[j]))
}
simdata <- as.data.frame(seg)
rm(p, M, seg, segs, j)
#plot(data[, 1])

## Not run:
## CE with the four parameter beta distribution with BIC as the selection criterion ##
obj1 <- CE.NB(simdata, distyp = 1, penalty = BIC, parallel = TRUE) # Parallel computation
obj1
profilePlot(obj1, simdata) # To obtain the mean profile plot

## CE with truncated normal distribution with BIC as the selection criterion ##
obj2 <- CE.NB(simdata, distyp = 2, penalty = BIC, parallel = TRUE) # Parallel computation
obj2
profilePlot(obj1, simdata) # To obtain the mean profile plot

## End(Not run)
```

Description

Performs calculations to estimate the break-point locations when their initial values are given. Negative binomial distribution is used to model the over-dispersed discrete (count) data. This function supports for the simulation of break-point locations in the CE algorithm based on the four parameter beta distribution or truncated normal distribution. User can select either BIC or AIC to select the optimal number of break-points.
Usage

CE.NB.Init(data, init.locs, eps = 0.01, rho = 0.05, M = 200, h = 5, a = 0.8, b = 0.8, distyp = 1, penalty = "BIC", var.init = 1e+05, parallel = FALSE)

Arguments

data data to be analysed. A single column array or a dataframe.
init.locs Initial break-point locations.
eps the cut-off value for the stopping criterion in the CE method. Default value is 0.01.
rho the fraction which is used to obtain the best performing set of sample solutions (i.e., elite sample). Default value is 0.05.
M sample size to be used in simulating the locations of break-points. Default value is 200.
h minimum aberration width. Default is 5.
a a smoothing parameter value. It is used in the four parameter beta distribution to smooth both shape parameters. When simulating from the truncated normal distribution, this value is used to smooth the estimates of the mean values. Default is 0.8.
b a smoothing parameter value. It is used in the truncated normal distribution to smooth the estimates of the standard deviation. Default is 0.8.
distyp distribution to simulate break-point locations. Options: 1 = four parameter beta distribution, 2 = truncated normal distribution. Default is 1.
penalty User can select either BIC or AIC to obtain the number of break-points. Options: "BIC", "AIC". Default is "BIC".
var.init Initial variance value to facilitate the search process. Default is 100000.
parallel A logical argument specifying if parallel computation should be carried-out (TRUE) or not (FALSE). By default it is set as 'FALSE'. In WINDOWS OS systems "snow" functionalities are used, whereas in Unix/Linux/MAC OSX "multicore" functionalities are used to carryout parallel computations with the maximum number of cores available.

Details

The negative binomial (NB) distribution is used to model the discrete (count) data. NB model is preferred over the Poission model when over-dispersion is observed in the count data. A performance function score (BIC or AIC) is calculated for each of the solutions generated by the statistical distribution (four parameter beta distribution or truncated normal distribution) with respect to the user provided initial locations. Finally, a list containing a vector of break-point locations, number of break-points, BIC/AIC values and log-likelihood value is returned in the console.

Value

A list is returned with following items:

No.BPs The number of break-points
BP.Loc  A vector of break-point locations
BIC/AIC  BIC/AIC value
ll      Loglikelihood of the optimal solution

Author(s)

Priyadarshana, W.J.R.M. <mjayawardana@swin.edu.au>

References


See Also

CE.NB for CE with Negative binomial,
CE.ZINB for CE with zero-inflated negative binomial,
CE.ZINB.Init for CE with zero-inflated negative binomial with initial locations,
profilePlot to obtain mean profile plot.

Examples

```r
#### Simulated data example ####
segs <- 6  # Number of segments
M <- c(1500, 2200, 800, 2500, 1000, 2000)  # Segment width
true.locations <- c(1501, 3701, 4501, 7001, 8001)  # True break-point locations
seg <- NULL
p <- c(0.45, 0.25, 0.4, 0.2, 0.3, 0.6)  # Specification of p's for each segment
for(j in 1:segs){
  seg <- c(seg, rnbinom(M[j], size =10, prob = p[j]))
}
simdata <- as.data.frame(seg)
rm(p, M, seg, segs, j)
#plot(data[, 1])

## Not run:
## CE with the four parameter beta distribution with BIC as the selection criterion ##

##Specification of initial locations
```
init.locations <- c(1400, 3400, 4650, 7100, 8200)

obj1 <- CE.NB.Init(simdata, init.locs = init.locations, distyp = 1, penalty = BIC, parallel = TRUE)

profilePlot(obj1, simdata) # To obtain the mean profile plot

## CE with truncated normal distribution with BIC as the selection criterion ##

obj2 <- CE.NB.Init(simdata, init.locs = init.locations, distyp = 2, penalty = BIC, parallel = TRUE)

profilePlot(obj1, simdata) # To obtain the mean profile plot

## End(Not run)##

---

**CE.Normal.Init.Mean**  
*Multiple break-point detection via the CE method for continuous data with initial locations (mean levels)*

**Description**
Performing calculations to estimate the break-point locations when their initial values are given. Normal distribution is used to model the observed continuous data. Across the segments standard deviation is assumed to be the same. This function supports for the simulation of break-point locations based on the four parameter beta distribution or truncated normal distribution. User can select from the modified BIC (mBIC) proposed by Zhang and Siegmund (2007), BIC or AIC to obtain the optimal number of break-points.

**Usage**

```r
CE.Normal.Init.Mean(data, init.locs, eps = 0.01, rho = 0.05, M = 200, h = 5, a = 0.8, 
b = 0.8, distyp = 1, penalty = "mBIC", var.init = 1e+05, parallel = FALSE)
```

**Arguments**
- **data**  
  data to be analysed. A single column array or a dataframe.
- **init.locs**  
  Initial break-point locations.
- **eps**  
  the cut-off value for the stopping criterion in the CE method. Default value is 0.01.
- **rho**  
  the fraction which is used to obtain the best performing set of sample solutions (i.e., elite sample). Default value is 0.05.
- **M**  
  sample size to be used in simulating the locations of break-points. Default value is 200.
- **h**  
  minimum aberration width. Default is 5.
a smoothing parameter value. It is used in the four parameter beta distribution to
smooth both shape parameters. When simulating from the truncated normal dis-
tribution, this value is used to smooth the estimates of the mean values. Default
is 0.8.

b smoothing parameter value. It is used in the truncated normal distribution to
smooth the estimates of the standard deviation. Default is 0.8.

distyp distribution to simulate break-point locations. Options: 1 = four parameter beta
distribution, 2 = truncated normal distribution. Default is 1.

penalty User can select from mBIC, BIC or AIC to obtain the optimal number of break-
points. Options: "mBIC", "BIC" and "AIC". Default is "mBIC".

var.init Initial variance value to facilitate the search process. Default is 100000.

parallel A logical argument specifying if parallel computation should be carried-out
(TRUE) or not (FALSE). By default it is set as ‘FALSE’. In WINDOWS OS
systems "snow" functionalities are used, whereas in Unix/Linux/MAC OSX
"multicore" functionalities are used to carryout parallel computations with the
maximum number of cores available.

Details

The normal distribution is used to model the continuous data. A performance function score
(mBIC/BIC/AIC) is calculated for each of the solutions generated by the statistical distribution (four
parameter beta distribution or truncated normal distribution), which is used to simulate break-points
from the user provided initial locations. The solution that maximizes the selection criteria with re-
spect to the number of break-points is reported as the optimal solution. Finally, a list containing a
vector of break-point locations, number of break-points, mBIC/BIC/AIC values and log-likelihood
value is returned in the console.

Value

A list is returned with following items:

- **noNbps** The number of break-points
- **bpNloc** A vector of break-point locations
- **mbic/bic/aic** mBIC/BIC/AIC value
- **ll** Loglikelihood of the optimal solution

Author(s)

Priyadarshana, W.J.R.M. <mjayawardana@swin.edu.au>

References

Data via the Cross-Entropy Method, IEEE/ACM Transactions on Computational Biology and Bioin-

ing Multiple Change-Points in DNA Count Data, In Proc. of the IEEE Conference on Evolutionary
Computation (CEC), 1020-1027, DOI: 10.1109/CEC.2012.6256470.


See Also
CE.Normal.Mean for CE with normal,
CE.Normal.MeanVar for CE with normal to detect break-points in both mean and variance,
CE.Normal.Init.MeanVar for CE with normal to detect break-points in both mean and variance with initial locations,
profilePlot to obtain mean profile plot.

Examples
## Not run:
simdata <- as.data.frame(c(rnorm(200,100,5),rnorm(100,300,5),rnorm(300,150,5)))

## CE with four parameter beta distribution with mBIC as the selection criterion
##
## obj1 <- CE.Normal.Init.Mean(simdata, init.locs = c(150, 380), distyp = 1, parallel =TRUE)
## profilePlot(obj1, simdata)

## CE with truncated normal distribution with mBIC as the selection criterion
##
## obj2 <- CE.Normal.Init.Mean(simdata, init.locs = c(150, 380), distyp = 2, parallel =TRUE)
## profilePlot(obj2, simdata)

## End(Not run)
Arguments

- **data**: data to be analysed. A single column array or a dataframe.
- **init.locs**: Initial break-point locations.
- **eps**: the cut-off value for the stopping criterion in the CE method. Default value is 0.01.
- **rho**: the fraction which is used to obtain the best performing set of sample solutions (i.e., elite sample). Default value is 0.05.
- **M**: sample size to be used in simulating the locations of break-points. Default value is 200.
- **h**: minimum aberration width. Default is 5.
- **a**: a smoothing parameter value. It is used in the four parameter beta distribution to smooth both shape parameters. When simulating from the truncated normal distribution, this value is used to smooth the estimates of the mean values. Default is 0.8.
- **b**: a smoothing parameter value. It is used in the truncated normal distribution to smooth the estimates of the standard deviation. Default is 0.8.
- **distyp**: distribution to simulate break-point locations. Options: 1 = four parameter beta distribution, 2 = truncated normal distribution. Default is 1.
- **penalty**: User can select either from BIC or AIC to obtain the optimal number of break-points. Options: "BIC" and "AIC". Default is "BIC".
- **var.init**: Initial variance value to facilitate the search process. Default is 100000.
- **parallel**: A logical argument specifying if parallel computation should be carried-out (TRUE) or not (FALSE). By default it is set as ‘FALSE’. In WINDOWS OS systems "snow" functionalities are used, whereas in Unix/Linux/MAC OSX "multicore" functionalities are used to carry out parallel computations with the maximum number of cores available.

Details

The normal distribution is used to model the continuous data. A performance function score (BIC/AIC) is calculated for each of the solutions generated by the statistical distribution (four parameter beta distribution or truncated normal distribution), which is used to simulate break-points from the user provided initial locations. Changes in both mean and variances are estimated. The solution that maximizes the selection criteria with respect to the number of break-points is reported as the optimal solution. Finally, a list containing a vector of break-point locations, number of break-points, BIC/AIC values and log-likelihood value is returned in the console.

Value

A list is returned with following items:

- **No. BPs**: The number of break-points
- **BP. Loc**: A vector of break-point locations
- **BIC/AIC**: BIC/AIC value
- **ll**: Loglikelihood of the optimal solution
Author(s)

Priyadarshana, W.J.R.M. <mjayawardana@swin.edu.au>

References


See Also

CE.Normal.Init.Mean for CE with normal with initial locations,
CE.Normal.Mean for CE with normal to detect break-points in mean levels,
CE.Normal.MeanVar for CE with normal to detect break-points in both mean and variance,
profilePlot to obtain mean profile plot.

Examples

```r
## Not run:
simdata <- as.data.frame(c(rnorm(200,100,5),rnorm(1000,160,8),rnorm(300,120,10)))
initial.locs <- c(225, 1300)

## CE with four parameter beta distribution with BIC as the selection criterion ##
obj1 <- CE.Normal.Init.MeanVar(simdata, init.locs = initial.locs, distyp = 1, parallel =TRUE)
profilePlot(obj1, simdata)

## CE with truncated normal distribution with BIC as the selection criterion ##
obj2 <- CE.Normal.Init.MeanVar(simdata, init.locs = initial.locs, distyp = 2, parallel =TRUE)
profilePlot(obj2, simdata)

## End(Not run)
```
Multiple Break-point Detection via the CE Method for Continuous Data (Mean levels)

Description
This function performs calculations to estimate both the number of break-points and their corresponding locations of continuous measurements with the CE method. The normal distribution is used to model the observed continuous data. Across the segments standard deviation is assumed to be the same. This function supports for the simulation of break-point locations based on the four parameter beta distribution or truncated normal distribution. User can select from the modified BIC (mBIC) proposed by Zhang and Siegmund (2007), BIC or AIC to obtain the optimal number of break-points.

Usage
CE.Normal.Mean(data, Nmax = 10, eps = 0.01, rho = 0.05, M = 200, h = 5, a = 0.8, b = 0.8, distyp = 1, penalty = "mBIC", parallel = FALSE)

Arguments
- `data`: data to be analysed. A single column array or a dataframe.
- `Nmax`: maximum number of break-points. Default value is 10.
- `eps`: the cut-off value for the stopping criterion in the CE method. Default value is 0.01.
- `rho`: the fraction which is used to obtain the best performing set of sample solutions (i.e., elite sample). Default value is 0.05.
- `M`: sample size to be used in simulating the locations of break-points. Default value is 200.
- `h`: minimum aberration width. Default is 5.
- `a`: a smoothing parameter value. It is used in the four parameter beta distribution to smooth both shape parameters. When simulating from the truncated normal distribution, this value is used to smooth the estimates of the mean values. Default is 0.8.
- `b`: a smoothing parameter value. It is used in the truncated normal distribution to smooth the estimates of the standard deviation. Default is 0.8.
- `distyp`: distributions to simulate break-point locations. Options: 1 = four parameter beta distribution, 2 = truncated normal distribution. Default is 1.
- `penalty`: User can select from mBIC, BIC or AIC to obtain the optimal number of break-points. Options: "mBIC", "BIC" and "AIC". Default is "mBIC".
- `parallel`: A logical argument specifying if parallel computation should be carried-out (TRUE) or not (FALSE). By default it is set as 'FALSE'. In WINDOWS OS systems "snow" functionalities are used, whereas in Unix/Linux/MAC OSX "multicore" functionalities are used to carryout parallel computations with the maximum number of cores available.
Details

The normal distribution is used to model the continuous data. A performance function score (mBIC/BIC/AIC) is calculated for each of the solutions generated by the statistical distribution (four parameter beta distribution or truncated normal distribution), which is used to simulate break-points from no break-point to the user provided maximum number of break-points. The solution that maximizes the selection criteria with respect to the number of break-points is reported as the optimal solution. Finally, a list containing a vector of break-point locations, number of break-points, mBIC/BIC/AIC values and log-likelihood value is returned in the console.

Value

A list is returned with following items:

- No.BPs: The number of break-points
- BP.Loc: A vector of break-point locations
- mBIC/BIC/AIC: mBIC/BIC/AIC value
- ll: Loglikelihood of the optimal solution

Author(s)

Priyadarshana, W.J.R.M. <mjayawardana@swin.edu.au>

References


See Also

- CE_Normal_Init_Mean for CE with normal with initial locations,
- CE_Normal_MeanVar for CE with normal to detect break-points in both mean and variance,
- CE_Normal_Init_MeanVar for CE with normal to detect break-points in both mean and variance with initial locations,
- profilePlot to obtain mean profile plot.
Examples

data(ch1.GM03563)
## Not run:
## CE with four parameter beta distribution with mBIC as the selection criterion
##
obj1 <- CE.Normal.MeanVar(ch1.GM03563, distyp = 1, penalty = "mBIC", parallel = TRUE)
profilePlot(obj1, simdata)

## CE with truncated normal distribution with mBIC as the selection criterion
##
obj2 <- CE.Normal.MeanVar(ch1.GM03563, distyp = 2, penalty = "mBIC", parallel = TRUE)
profilePlot(obj2, simdata)

## End(Not run)

CE.Normal.MeanVar  Multiple break-point detection via the CE method for continuous data
(both mean and variance changes)

Description

This function performs calculations to estimate both the number of break-points and their corresponding locations of continuous measurements with the CE method. The normal distribution is used to model the observed continuous data. This function supports for the simulation of break-point locations based on the four parameter beta distribution or truncated normal distribution. User can select either from the general BIC or AIC to obtain the optimal number of break-points.

Usage

CE.Normal.MeanVar(data, Nmax = 10, eps = 0.01, rho = 0.05, M = 200, h = 5, a = 0.8, b = 0.8, distyp = 1, penalty = "BIC", parallel = FALSE)

Arguments

data  data to be analysed. A single column array or a dataframe.
Nmax  maximum number of break-points. Default value is 10.
eps  the cut-off value for the stopping criterion in the CE method. Default value is 0.01.
rho  the fraction which is used to obtain the best performing set of sample solutions (i.e., elite sample). Default value is 0.05.
M  sample size to be used in simulating the locations of break-points. Default value is 200.
h  minimum aberration width. Default is 5.
a  a smoothing parameter value. It is used in the four parameter beta distribution to smooth both shape parameters. When simulating from the truncated normal distribution, this value is used to smooth the estimates of the mean values. Default is 0.8.
b

da smoothing parameter value. It is used in the truncated normal distribution to
smooth the estimates of the standard deviation. Default is 0.8.

distyp

distributions to simulate break-point locations. Options: 1 = four parameter beta
distribution, 2 = truncated normal distribution. Default is 1.

penalty

User can select from BIC or AIC to obtain the optimal number of break-points.
Options: "BIC" and "AIC". Default is "BIC".

parallel

A logical argument specifying if parallel computation should be carried-out
(TRUE) or not (FALSE). By default it is set as ‘FALSE’. In WINDOWS OS
systems "snow" functionalities are used, whereas in Unix/Linux/MAC OSX
"multicore" functionalities are used to carry out parallel computations with the
maximum number of cores available.

Details

The normal distribution is used to model the continuous data. A performance function score
(BIC/AIC) is calculated for each of the solutions generated by the statistical distribution (four pa-
rameter beta distribution or truncated normal distribution), which is used to simulate break-points
from no break-point to the user provided maximum number of break-points. Changes in both mean
and variance are estimated. The solution that maximizes the selection criteria with respect to the
number of break-points is reported as the optimal solution. Finally, a list containing a vector of
break-point locations, number of break-points, BIC/AIC values and log-likelihood value is returned
in the console.

Value

A list is returned with following items:

No.BPs The number of break-points
BP.Loc A vector of break-point locations
BIC/AIC BIC/AIC value
ll Loglikelihood of the optimal solution

Author(s)

Priyadarshana, W.J.R.M. <mjayawardana@swin.edu.au>

References

Data via the Cross-Entropy Method, IEEE/ACM Transactions on Computational Biology and Bioin-

ing Multiple Change-Points in DNA Count Data, In Proc. of the IEEE Conference on Evolutionary
Computation (CEC), 1020-1027, DOI: 10.1109/CEC.2012.6256470.

binatorial Optimization, Monte-Carlo Simulation and Machine Learning. Springer-Verlag, New
York.

See Also

CE.Normal_Init_Mean for CE with normal with initial locations,
CE.Normal_Mean for CE with normal to detect break-points in mean levels,
CE.Normal_Init_MeanVar for CE with normal to detect break-points in both mean and variance with initial locations,
profilePlot to obtain mean profile plot.

Examples

```r
## Not run:
simdata <- as.data.frame(c(rnorm(200,100,5),rnorm(1000,160,8),rnorm(300,120,10)))

## CE with four parameter beta distribution with BIC as the selection criterion  ##
objc <- CE.Normal_MeanVar(simdata, distyp = 1, penalty = "BIC", parallel =TRUE)
profilePlot(objc, simdata)

## CE with truncated normal distribution with BIC as the selection criterion  ##
objc2 <- CE.Normal_MeanVar(simdata, distyp = 2, penalty = "BIC", parallel =TRUE)
profilePlot(objc2, simdata)

## End(Not run)
```

Description

Performs calculations to estimate both the number of break-points and their corresponding locations of discrete measurements with the CE method. Zero-inflated negative binomial distribution is used to model the excess zero observations and to model over-dispersion in the observed discrete (count) data. This function supports for the simulation of break-point locations in the CE algorithm based on the four parameter beta distribution and truncated normal distribution. The general BIC or AIC can be used to select the optimal number of break-points.

Usage

```r
CE.ZINB(data, Nmax = 10, eps = 0.01, rho = 0.05, M = 200, h = 5, a = 0.8, b = 0.8,
distyp = 1, penalty = "BIC", parallel = FALSE)
```
Arguments

- **data**: data to be analysed. A single column array or a dataframe.
- **nmax**: maximum number of break-points. Default value is 10.
- **eps**: the cut-off value for the stopping criterion in the CE method. Default value is 0.01.
- **rho**: the fraction which is used to obtain the best performing set of sample solutions (i.e., elite sample). Default value is 0.05.
- **M**: sample size to be used in simulating the locations of break-points. Default value is 200.
- **h**: minimum aberration width. Default is 5.
- **a**: a smoothing parameter value. It is used in the four parameter beta distribution to smooth both shape parameters. When simulating from the truncated normal distribution, this value is used to smooth the estimates of the mean values. Default is 0.8.
- **b**: a smoothing parameter value. It is used in the truncated normal distribution to smooth the estimates of the standard deviation. Default is 0.8.
- **distyp**: distribution to simulate break-point locations. Options: 1 = four parameter beta distribution, 2 = truncated normal distribution. Default is 1.
- **penalty**: User can select either BIC or AIC to obtain the number of break-points. Options: "BIC", "AIC". Default is "BIC".
- **parallel**: A logical argument specifying if parallel computation should be carried-out (TRUE) or not (FALSE). By default it is set as ‘FALSE’. In WINDOWS OS systems "snow" functionalities are used, whereas in Unix/Linux/MAC OSX "multicore" functionalities are used to carryout parallel computations with the maximum number of cores available.

Details

Zero-inflated negative binomial (ZINB) distribution is used to model the discrete (count) data. ZINB model is preferred over the NB model when both excess zero values and over-dispersion observed in the count data. A performance function score (BIC) is calculated for each of the solutions generated by the statistical distribution (four parameter beta distribution or truncated normal distribution), which is used to simulate break-points from no break-point to the user provided maximum number of break-points. The solution that minimizes the BIC/AIC with respect to the number of break-points is reported as the optimal solution. Finally, a list containing a vector of break-point, BIC/AIC values and log-likelihood value is returned in the console.

Value

A list is returned with following items:

- **No.BPs**: The number of break-points
- **BP.Loc**: A vector of break-point locations
- **BIC/AIC**: BIC/AIC value
- **ll**: Loglikelihood of the optimal solution
Author(s)

Priyadarshana, W.J.R.M. <mjayawardana@swin.edu.au>

References


See Also

`CE.NB` for CE with negative binomial,
`CE.NB.Init` for CE with negative binomial with initial locations,
`CE.ZINB.Init` for CE with zero-inflated negative binomial with initial locations,
`profilePlot` to obtain mean profile plot.

Examples

```r
### Simulated data example ###
# gamlss R package is used to simulate data from the ZINB.

## Not run:
library(gamlss)
segs <- 6 # Number of segments
M <- c(1500, 2200, 800, 2500, 1000, 2000) # Segment width
true.locations <- c(1501, 3701, 4501, 7001, 8001) # True break-point locations
seg <- NULL
p <- c(0.6, 0.1, 0.3, 0.05, 0.2, 0.4) # Specification of p's on each segment
sigma.val <- c(1, 2, 3, 4, 5, 6) # Specification of sigma values

for(j in 1:segs){
  seg <- c(seg, rZINBI(M[j], mu = 300, sigma = sigma.val[j], nu = p[j]))
}

simdata <- as.data.frame(seg)
rm(p, M, seg, segs, j, sigma.val)
#plot(data[, 1])

## CE with the four parameter beta distribution with BIC as the selection criterion ##
obj1 <- CE.ZINB(simdata, distyp = 1, penalty = BIC, parallel = TRUE) # Parallel computation
```
obj1

profilePlot(obj1, simdata) # To obtain the mean profile plot

## CE with truncated normal distribution with BIC as the selection criterion ##

obj2 <- CE.ZINB(simdata, distyp = 2, penalty = BIC, parallel = TRUE) # Parallel computation

profilePlot(obj2, simdata) # To obtain the mean profile plot

## End(Not run)##

---

**CE.ZINB.Init**

*Multiple Break-point Detection via the CE Method with Zero-Inflated Negative Binomial Distribution with initial locations*

---

**Description**

Performs calculations to estimate the break-point locations when their initial values are given. Zero-inflated negative binomial distribution is used to model the excess zero observations and to model over-dispersesn in the observed discrete (count) data. This function supports for the simulation of break-point locations in the CE algorithm based on the four parameter beta distribution and truncated normal distribution. The general BIC or AIC can be used to select the optimal number of break-points.

**Usage**

`CE.ZINB.Init(data, init.locs, eps = 0.01, rho = 0.05, M = 200, h = 5, a = 0.8, b = 0.8, distyp = 1, penalty = "BIC", var.init = 1e+05, parallel = FALSE)`

**Arguments**

- **data**: data to be analysed. A single column array or a dataframe.
- **init.locs**: Initial break-point locations.
- **eps**: the cut-off value for the stopping criterion in the CE method. Default value is 0.01.
- **rho**: the fraction which is used to obtain the best performing set of sample solutions (i.e., elite sample). Default value is 0.05.
- **M**: sample size to be used in simulating the locations of break-points. Default value is 200.
- **h**: minimum aberration width. Default is 5.
- **a**: a smoothing parameter value. It is used in the four parameter beta distribution to smooth both shape parameters. When simulating from the truncated normal distribution, this value is used to smooth the estimates of the mean values. Default is 0.8.
a smoothing parameter value. It is used in the truncated normal distribution to smooth the estimates of the standard deviation. Default is 0.8.

distyp distribution to simulate break-point locations. Options: 1 = four parameter beta distribution. 2 = truncated normal distribution. Default is 1.

penalty User can select either BIC or AIC to obtain the number of break-points. Options: "BIC", "AIC". Default is "BIC".

var.init Initial variance value to facilitate the search process. Default is 100000.

parallel A logical argument specifying if parallel computation should be carried-out (TRUE) or not (FALSE). By default it is set as ‘FALSE’. In WINDOWS OS systems "snow" functionalities are used, whereas in Unix/Linux/MAC OSX "multicore" functionalities are used to carryout parallel computations with the maximum number of cores available.

Details

Zero-inflated negative binomial (ZINB) distribution is used to model the discrete (count) data. ZINB model is preferred over the NB model when both excess zero values and over-dispersion observed in the count data. A performance function score (BIC) is calculated for each of the solutions generated by the statistical distribution (four parameter beta distribution or truncated normal distribution), which is used to simulate break-points when the initial locations are provided. Finally, a list containing a vector of break-point locations, number of break-points, BIC/AIC values and log-likelihood value is returned in the console.

Value

A list is returned with following items:

- No.BPs The number of break-points
- BP.Loc A vector of break-point locations
- BIC/AIC BIC/AIC value
- ll Loglikelihood of the optimal solution

Author(s)

Priyadarshana, W.J.R.M. <mjayawardana@swin.edu.au>

References


See Also

CE.NB for CE with negative binomial,
CE.NB.Init for CE with negative binomial with initial locations,
CE.ZINB for CE with zero-inflated negative binomial,
profilePlot to obtain mean profile plot.

Examples

### Simulated data example ###
# gamlss R package is used to simulate data from the ZINB.

```
## Not run:
library(gamlss)
segs <- 6 # Number of segments
M <- c(1500, 2200, 800, 2500, 1000, 2000) # Segment width
true.locations <- c(1501, 3701, 4501, 7001, 8001) # True break-point locations
seg <- NULL
p <- c(0.6, 0.1, 0.3, 0.05, 0.2, 0.4) # Specification of p's on each segment
sigma.val <- c(1,2,3,4,5,6) # Specification of sigma values

for(j in 1:segs){
  seg <- c(seg, rZINBI(M[j], mu = 300, sigma = sigma.val[j], nu = p[j]))
}

simdata <- as.data.frame(seg)
rm(p, M, seg, segs, j, sigma.val)
#plot(data[, 1])

## CE with the four parameter beta distribution with BIC as the selection criterion ##
init.loci <- c(1400, 3400, 4650, 7100, 8200)
obj1 <- CE.ZINB.Init(simdata, init.locs = init.loci, distyp = 1, penalty = BIC, parallel = TRUE)
obj1
profilePlot(obj1, simdata) # To obtain the mean profile plot

## CE with truncated normal distribution with BIC as the selection criterion ##
obj2 <- CE.ZINB.Init(simdata, init.locs = init.loci, distyp = 2, penalty = BIC, parallel = TRUE)
obj2
profilePlot(obj2, simdata) # To obtain the mean profile plot

## End(Not run)```
**Description**

Chromosome 1 of cell line GM03563

**Usage**

data("ch1.GM03563")

**Format**

A single column data frame with 135 observations corresponds to chromosome 1 of cell line GM03563.

$log2ratio$ normalized average of the log base 2 test over reference ratio data

**Details**

This data set is extracted from a single experiments on 15 fibroblast cell lines with each array containing over 2000 (mapped) BACs spotted in triplicate discussed in Snijders et al.(2001). Data corresponds to the chromosome 1 of cell line GM03563.

**References**


**Examples**

data(ch1.GM03563)

```r
## Not run:
## CE with four parameter beta distribution ##
obj1 <- CE.Normal.Mean(ch1.GM03563, distyp = 1, parallel =TRUE) profilePlot(obj1, ch1.GM03563)

## CE with truncated normal distribution ##
obj2 <- CE.Normal.Mean(ch1.GM03563, distyp = 2, parallel =TRUE) profilePlot(obj2, ch1.GM03563)

## End(Not run)
```
profilePlot

---

**Description**

Plotting function to obtain mean profile plot of the testing dataset based on the estimates of the break-points. An R object created from the CE.Normal, CE.NB or CE.ZINB is required. User can alter the axis names.

**Usage**

```r
profilePlot(obj, data, x.label = "Data Sequence", y.label = "Value")
```

**Arguments**

- `obj` R object created from CE.Normal, CE.NB or CE.ZINB.
- `data` data to be analysed. A single column array or a dataframe.
- `x.label` x axis label. Default is "Data Sequence".
- `y.label` y axis label. Default is "Value".

**Author(s)**

Priyadarshana, W.J.R.M. <mjayawardana@swin.edu.au>

**See Also**

- `CE.Normal.Mean`
- `CE.NB`
- `CE.ZINB`

**Examples**

```r
data(ch1.GM03563)
## Not run:
## CE with four parameter beta distribution ##
obj1 <- CE.Normal.Mean(ch1.GM03563, distyp = 1, penalty = "mBIC", parallel =TRUE)
profilePlot(obj1)

## CE with truncated normal distribution ##
obj2 <- CE.Normal.Mean(ch1.GM03563, distyp = 2, penalty = "mBIC", parallel =TRUE)
profilePlot(obj2)

## End(Not run)
```
Index

*Topic data
  ch1.GM03563, 23

breakpoint (breakpoint-package), 2
breakpoint-package, 2

CE.NB, 3, 7, 19, 22, 24
CE.NB.Init, 5, 5, 19, 22
CE.Normal.Init.Mean, 8, 12, 14, 17
CE.Normal.Init.MeanVar, 10, 10, 14, 17
CE.Normal.Mean, 10, 12, 13, 17, 24
CE.Normal.MeanVar, 10, 12, 14, 15
CE.ZINB, 5, 7, 17, 22, 24
CE.ZINB.Init, 5, 7, 19, 20
ch1.GM03563, 23

profilePlot, 5, 7, 10, 12, 14, 17, 19, 22, 24