Package ‘crops’

October 12, 2022

Type      Package
Title     Changepoints for a Range of Penalties (CROPS)
Version   1.0.3
Date      2022-08-05
Maintainer Daniel Grose <dan.grose@lancaster.ac.uk>
Description Implements the Changepoints for a Range of Penalties (CROPS) algorithm of Haynes et al. (2017) <doi:10.1080/10618600.2015.1116445> for finding all of the optimal segmentations for multiple penalty values over a continuous range.
License    GPL
Imports    sets, reshape, tidyverse, memoise, ggplot2, magrittr, methods, cowplot, tibble, Rdpack
Suggests   fpop, pacman
RdMacros   Rdpack
RoxygenNote 7.1.2
NeedsCompilation no
Author     Daniel Grose [aut, cre],
           Paul Fearnhead [aut],
           Idris Eckley [ctb]
Repository  CRAN
Date/Publication 2022-08-05 17:10:01 UTC

R topics documented:
crops         2
plot           4
print          5
segmentations  5
subset         6
summary        6
unique         7

Index  8
Providing a generic implementation of the crops (changepoints for a range of penalties) algorithm of Haynes et al. (2014) which efficiently searches a range of penalty values in multiple changepoint problems. The crops algorithm finds the optimal segmentations for a different number of segments without incurring as large a computational cost as solving the constrained optimisation problem for a range of values for the number of changepoints. To make the method generic, the user must provide a function that maps a penalty value to the results obtained by a penalised cost changepoint method, and formats these results in a specific way. This interface to the generic method is similar to that as used by the optimx package.

Usage

\[
crops(method, beta_min, beta_max, max_iterations = 20, \ldots)
\]

Arguments

- **method**: A function mapping a penalty value to the results obtained by a penalised cost changepoint method. The function must return a list containing the cost and a vector of changepoint locations corresponding to the optimal segmentation as determined by a penalised cost changepoint method.

- **beta_min**: A positive numeric value indicating the smallest penalty value to consider.

- **beta_max**: A positive numeric value indicating the maximum penalty value to consider.

- **max_iterations**: Positive non zero integer. Limits the maximum number of iterations of the crops algorithm to \(\text{max}\_\text{iterations}\). Default value is \(\text{max}\_\text{iterations}=20\).

- **...**: Additional parameters to pass to the underlying changepoint method if required.

Value

An instance of an S4 class of type `crops.class`.

References


# generate some simple data
set.seed(1)
N <- 100
data.vec <- c(rnorm(N), rnorm(N, 2), rnorm(N))

# example one - calling fpop via crops using global scope
# need the fpop library
library(pacman)
p_load(fpop)

# create a function to wrap a call to fpop for use with crops
fpop.for.crops <- function(beta)
{
    # Note - this code is taken from the example in the fpop package
    fit <- Fpop(data.vec, beta)
    end.vec <- fit$t.est
    change.vec <- end.vec[-length(end.vec)]
    start.vec <- c(1, change.vec+1)
    segs.list <- list()
    for(seg.i in seq_along(start.vec))
    {
        start <- start.vec[seg.i]
        end <- end.vec[seg.i]
        seg.data <- data.vec[start:end]
        seg.mean <- mean(seg.data)
        segs.list[[seg.i]] <- data.frame(
            start, end,
            mean=seg.mean,
            seg.cost=sum((seg.data-seg.mean)^2))
    }
    segs <- do.call(rbind, segs.list)
    return(list(sum(segs$seg.cost),seg$end[-length(segs$end)]))
}

# now use this wrapper function with crops
res<-crops(fpop.for.crops,0.5*log(300),2.5*log(300))
# print summary of analysis
summary(res)
# summarise the segmentations
segmentations(res)
# visualise the segmentations
plot(res)
# overlay the data on the segmentations


df <- data.frame("x"=1:300,"y"=data.vec)
plot(res,df)

plot

Visualisation of data, costs, penalty values and changepoint locations.

Description

Plot methods for an S4 object returned by crops. The plot can also be combined with the original data if required.

Usage

## S4 method for signature 'crops.class,data.frame'
plot(x, y)

## S4 method for signature 'crops.class,missing'
plot(x)

Arguments

x  An instance of an S4 class produced by crops.
y  A dataframe containing the locations and values of the data points. The data plot is plotted below, and is aligned with, the changepoint locations.

Value

A ggplot object. Note - if no changepoints are detected in the penalty interval [beta_min,beta_max], then the value returned is NULL.

See Also

crops.

Examples

# see the crops example
# see the crops example
Pretty printing for crops results

Description
Pretty prints a summary of a crops result

Usage
```r
## S4 method for signature 'crops.class'
print(x)
```

Arguments
- `x`: An instance of an S4 class produced by `crops`.

See Also
- `crops`.

Examples
```r
# see the crops example
```

Summary of segmentations by penalty value

Description
Produces a summary of the segmentations for each penalty value in the form of a data frame.

Usage
```r
segmentations(object)
```

Arguments
- `object`: An instance of an S4 class produced by `crops`.

Value
A data frame containing the penalties, costs, penalised costs, and changepoint locations. Note - if no changepoints are detected in the penalty interval [beta_min,beta_max], then the value returned is NULL.
See Also
crops.

Examples

# see the crops example

subset  

<table>
<thead>
<tr>
<th>Subset crops results based on penalty values</th>
</tr>
</thead>
</table>

Description

Removes entries from a crops result that fall outside a specified range of penalty values. The subset function can be useful for simplifying plots and the details produced by segmentations.

Usage

## S4 method for signature 'crops.class'
subset(x, beta_min = 0, beta_max = Inf)

Arguments

- **x**: An instance of an S4 class produced by `crops`.
- **beta_min**: A positive numeric value specifying the minimum penalty value for entries in the crops result. Default value is 0.
- **beta_max**: A positive numeric value specifying the maximum penalty value for entries in the crops result. Default value is Inf.

Value

An instance of the S4 class type `crops.class`. This is the same type as produced by the `crops` function.

summary  

<table>
<thead>
<tr>
<th>Summary of crops result</th>
</tr>
</thead>
</table>

Description

Prints a short summary of a crops result.

Usage

## S4 method for signature 'crops.class'
summary(object)
Remove duplicate entries from a crops result. A duplicate entry is one having the same number of changepoints as another entry. Note that the changepoint locations and the associated penalty and cost values are not taken into consideration. The `unique` function can be useful for simplifying plots and the details produced by segmentations.

**Usage**

```r
## S4 method for signature 'crops.class'
unique(x)
```

**Arguments**

- `x`: An instance of an S4 class produced by `crops`.

**Value**

An instance of the S4 class type `crops.class`. This is the same type as produced by the `crops` function.
Index

crops, 2, 4–7
plot, 4
plot, crops.class, data.frame-method (plot), 4
plot, crops.class, missing-method (plot), 4
print, 5
print, crops.class-method (print), 5
segmentations, 5
segmentations, crops.class-method (segmentations), 5
subset, 6
subset, crops.class-method (subset), 6
summary, 6
summary, crops.class-method (summary), 6
unique, 7
unique, crops.class-method (unique), 7