Package 'ocp'

April 21, 2019

Type Package

Title Bayesian Online Changepoint Detection

Version 0.1.1

Author Andrea Pagotto

Maintainer Andrea Pagotto <ajpagotto@gmail.com>

Description Implements the Bayesian online changepoint detection method by Adams and MacKay (2007) <arXiv:0710.3742> for univariate or multivariate data. Gaussian and Poisson probability models are implemented. Provides post-processing functions with alternative ways to extract changepoints.

Encoding UTF-8

License GPL-3

LazyData true

RoxygenNote 6.1.0.9000

Imports grid (>= 3.4.0), graphics (>= 3.4.0), grDevices (>= 3.4.0)

Depends R (>= 3.4.0)

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-04-21 15:00:03 UTC

R topics documented:

`ocp-package` *Bayesian Online Changepoint Detection for Multivariate Data*

Description

Provides an implementation of Bayesian online changepoint detection. Handles multivariate and missing data. Computes the set of changepoints with highest probability in an online way (updating the results with each incoming point). Also provides post-processing functions with alternative ways to extract changepoints.

Author(s)

Pagotto, Andrea

`const_hazard` *Constant hazard function*

Description

Hazard function for use with gaussian underlying distribution.

Usage

```
const_hazard(r, lambda)
```

Arguments

`r`	The current R vector length.
`lambda`	The parameter for the hazard function.

Value

A vector of the hazard function for the length of the current R vector.

Examples

```
H<- const_hazard(10, 1/100)
```

`findCPprobs` *Find Set of Changepoints with Highest probability*

Description

This function calculates the changepoints with highest probability in the online algorithm to take in the current probabilities at time t in the form of a list of lists. It will not calculate the result at every possible end point, because this will be done in the main loop of online cpd as it iterates: the probmaxes and cps list will be returned and passed into the function again each time.

Usage

```
findCPprobs(currrunprobs, probmaxes, logprobcpstrunc, Rlength, t,
  minsep = 3, maxsep = 90, ppres = FALSE)
```

Arguments

`currrunprobs`	The current most recently calculated "R" vector, of run length probabilities (sums to 1).
`probmaxes`	The probabilities of the set of changepoints with the highest probability for each preceding time point.
`logprobcpstrunc`	The set of changepoints with the highest probability for each previous time point.
`Rlength`	The length of the current R vector, to use in case it was truncated.
`t`	The current time point.
`minsep`	The minimum distance of separation allowed for eligible changepoint locations to be included in the list of changepoints with the highest probability.
`maxsep`	The maximum distance of separation allowed for eligible changepoint locations to be included in the list of changepoints with the highest probability.
`ppres`	Set to true if wanting to return optional outputs, useful for plotting and inspecting the algorithm, but not necessary.

Value

Two lists needed for the use in calculating this changepoints for the next incoming time point: the vector of max probabilities for each time point (probmaxes), and the list of changepoints with the highest probability at each time point (changepoints: a list of lists). It also returns ppresult: optional outputs, will be null if ppres=FALSE.

gamesdata *This is data to be included in the package*

Description

Data used in the LREC paper on the 2016 eurogames tweets. Includes a column with the counts of numbers of tweets. The columns present in the matrix at the three sentiment scores: "neg", "neu", and "pos".

Source

http://www.lrec-conf.org/proceedings/lrec2018/pdf/335.pdf

Examples

```
demo(eurogames)
```

gamesdatacounts *This is data to be included in the package*

Description

Data used in the LREC paper on the 2016 eurogames tweets. Includes a column with the counts of numbers of tweets. The columns present in the matrix at the three sentiment scores: "neg", "neu", and "pos", and an additional column for the total number of tweets: "counts"

Source

http://www.lrec-conf.org/proceedings/lrec2018/pdf/335.pdf

gaussianProb *Compute predictive probabilities based on Gaussian*

Description

Compute the probability of observing the current point, given the current parameters of the gaussian for each possible run length. Returns a vector of predictive probabilities from each possible run length, the parameters of the gaussian, the most likely mean of the current gaussian, and the current point.

Usage

```
gaussianProb(update_params0, update_paramsT, datapt, time, cps, missPts,
  Rlength, skippt = FALSE)
```

Arguments

update_params0	The initialization parameters, corresponding to predicting a changepoint (run length=0)
update_paramsT	The vectors of parameters corresponding to each possible run length, updated with each incoming data point
datapt	the current data point
time	the number of time points passed so far
cps	the current most likely list of changepoints
missPts	the method set to handle missing points
Rlength	the length of the current vector of possible run lengths
skippt	If the current point should be skipped in the updating because it was missing, and missPts was set to skip

Value

Returns a vector of predictive probabilities from each possible run length, the parameters of the gaussian, the most likely mean of the current gaussian, and the current point.

---

gaussian_init *Initialize vectors for gaussian probability functions*

Description

Takes in the desired initialization parameters, initializes the vectors needed for the gaussian probability function gaussian_update

Usage

```
gaussian_init(init_params = list(m = 0, k = 0.01, a = 0.01, b = 1e-04),
  dims)
```

Arguments

init_params	The list of parameters to be used for initialization
dims	the dimensionality of the data

Value

List of vectors to be used in the iteratively updating algorithm of parameters describing the underlying gaussian distribution of the data.

gaussian_update *Update the gaussian parameters*

Description

Updates the parameters of the gaussians based on each possible run length, after taking into consideration the most recent data point

Usage

```
gaussian_update(datapt, update_params0, update_paramsT, Rlength,
  skippt = FALSE)
```

Arguments

datapt	the current data point
update_params0	The initialization parameters, corresponding to predicting a changepoint (run length=0)
update_paramsT	The vectors of parameters corresponding to each possible run length, updated with each incoming data point
Rlength	the length of the current vector of possible run lengths
skippt	set to FALSE if not needing to accommodate skipping missed points during the update of parameters

Value

The list of the parameters for gaussians corresponding to each possible runlength up to the current data point. Lengths of vectors should correspond the length of the R vector ("run length vector")

initOCPD *Initialize ocpd object*

Description

This function initializes the ocpd object. It returns an ocpd object with no data, but matrixes and vectors set up to begin adding to throughout the running of the algorithm.

Usage

```
initOCPD(dims, init_params = list(list(m = 0, k = 0.01, a = 0.01, b =
  1e-04)), initProb = c(gaussian_init))
```

Arguments

dims	The dimensions calculated from the first input data points.
init_params	The list of params required to initialize the underlying distribution model.
initProb	The chosen type of underlying distribution.

Value

oCPD object initialized with initialization settings.

Examples

```
empty_ocpd<- initOCPD(1) # initialize bject with 1 dimensions
```

---

negbinpdf *Calculate Negative-binomial on vector of parameters*

Description

Computes the negative-binomial posterior predictive density from input parameter vectors corresponding to each possible run length for the current time point. Outputs a vector of probabilities for use in the accompanying poisson functions.

Usage

```
negbinpdf(x, a, b)
```

Arguments

x	the current data point
a	matrix of alpha params
b	matrix of beta params

Value

Matrix of negative binomial pdf values corresponding to each possible run length, for use in accompanying poisson probability functions.

onlineCPD *Bayesian Online Changepoint Detection*

Description

The main algorithm called "Bayesian Online Changepoint Detection". Input is data in form of a matrix and, optionally an existing ocp object to build on. Output is the list of changepoints and other values calculated during running the model.

Usage

```
onlineCPD(datapts, oCPD = NULL, missPts = "none",
  hazard_func = function(x, lambda) {     const_hazard(x, lambda = 100)
  }, probModel = list("g"), init_params = list(list(m = 0, k = 0.01, a
  = 0.01, b = 1e-04)), multivariate = FALSE, cpthreshold = 0.5,
  truncRlim = .Machine$double.xmin, minRlength = 1,
  maxRlength = 10^4, minsep = 1, maxsep = 10^4, timing = FALSE,
  getR = FALSE, optionalOutputs = FALSE, printupdates = FALSE)
```

Arguments

datapts	the input data in form of a matrix, where the rows correspond to each data point, and the columns correspond to each dimension.
oCPD	ocp object computed in a previous run of an algorithm. it can be built upon with the input data points, as long as the settings for both are the same.
missPts	This setting indicates how to deal with missing points (e.g. NA). The options are: "mean", "prev", "none", and a numeric value. If the data is multivariate. The numeric replacement value could either be a single value which would apply to all dimensions, or a vector of the same length as the number of dimensions of the data.
hazard_func	This setting allows choosing a hazard function, and also setting the constants within that function. For example, the default hazard function is: function(x, lambda)const_hazard(x, lambda=100) and the lambda can be set as appropriate.
probModel	This parameter is a function to be used to calculate the predictive probabilities and update the parameters of the model. The default setting uses a gaussian underlying distribution: "gaussian"
init_params	The parameters used to initialize the probability model. The default settings correspond to the input default gaussian model.
multivariate	This setting indicates if the incoming data is multivariate or univariate.
cpthreshold	Probability threshold for the method of extracting a list of all changepoints that have a run length probability higher than a specified value. The default is set to 0.5.
truncRlim	The probability threshold to begin truncating the R vector. The R vector is a vector of run-length probabilities. To prevent truncation, set this to 0. The defaults setting is 10^(-4) as suggested by the paper.

minRlength	The minimum size the run length probabilities vector must be before beginning to check for the truncation threshold.
maxRlength	The maximum size the R vector is allowed to be, before enforcing truncation to happen.
minsep	This setting constrains the possible changepoint locations considered in determining the optimal set of changepoints. It prevents considered changepoints that are closer together than the value of minsep. The default is 3.
maxsep	This setting constrains the possible changepoint locations considered in determining the optimal set of changepoints. It prevents considered changepoints that are closer farther apart than the value of maxsep. The default is 100.
timing	To print out times during the algorithm running, to track its progress, set this setting to true.
getR	To output the full R matrix, set this setting to TRUE. Outputting this matrix causes a major slow down in efficiency.
optionalOutputs	Output additional values calculated during running the algorithm, including a matrix containing all the input data, the predictive probability vectors at each step of the algorithm, and the vector of means at each step of the algorithm.
printupdates	This setting prints out updates on the progress of the algorithm if set to TRUE.

Value

An ocp object containing the main output: a list of changepoints from each time point, and many additional outputs: the number of time points, the initial settings of the algorithm, the current model parameters, the means from each time point, the most recently processed point, the most recently calculated vector of run length probabilities, and a vector of probabilities of changepoints at each time point.

Examples

```
simdatapts<- c(rnorm(n = 50), rnorm(n=50, 100))
ocpd1<- onlineCPD(simdatapts)
ocpd1$changepoint_lists # view the changepoint lists
```

---

plot.ocp *Plot Object*

Description

Plot ocpd object, to show the data and the R matrix probabilities.

Usage

```
## S3 method for class 'ocp'
plot(x, data = NULL, Rmat = NULL,
  graph_changepoints = TRUE, graph_probabilities = TRUE,
  showmaxes = TRUE, showmeans = TRUE, showcps = TRUE,
  showdata = TRUE, showRprobs = TRUE, cplistID = 3,
  main_title = "", trueCPs = NULL, showdataleg = TRUE,
  timepoints = NULL, timeunits = NULL, grey_digits = 4,
  varnames = NULL, ...)
```

Arguments

x	the ocp object to plot
data	optional input data to plot
Rmat	optional input Rmat to plot
graph_changepoints	set to TRUE to graph the changepoints
graph_probabilities	set TRUE to show R matrix graphed
showmaxes	set TRUE to show the maxes in each columns in the R matrix plot
showmeans	set TRUE to show the means on the changepoints plot
showcps	set TRUE to show the the locations of changepoints
showdata	set TRUE to show the actual data points
showRprobs	set TRUE to show the probabilities in the R matrix
cplistID	method of extracting the changepoints: either "colmaxes", "threshcps", or "maxCPs" stored in the "changepoints_list" in the ocpd object
main_title	The main title for both plots, e.g. "Eurogames Data"
trueCPs	input the true known changepoints for comparison
showdataleg	Set true to show legend for the data points, set to false if there are too many dimensions, legend will be crowded.
timepoints	List of timepoints to use as x-axis labels.
timeunits	Units to display for the timescale on the plot.
grey_digits	The limit of decimal places to keep in the probability before converting to an index in the grey-scale, controls amount of detail and darkness of the shading on the plot.
varnames	List of variable names to display in the legend.
...	(optional) additional arguments, ignored.

Examples

```
simdatapts<- c(rnorm(n = 50), rnorm(n=50, 100))
ocpd1<- onlineCPD(simdatapts, getR=TRUE)
plot(ocpd1) # basic plot
```

```
plot(ocpd1, data= simdatapts) # plot with the original data
plot(ocpd1, trueCPs = c(1, 51)) # plot with showing the true changepoints
plot(ocpd1, main_title="Example plot", showmaxes = FALSE) # not showing max probabilities
plot(ocpd1, graph_changepoints=FALSE) # not showing the changepoints plot
plot(ocpd1, graph_probabilities=FALSE) # not showing the R matrix
plot(ocpd1, showRprobs=FALSE, showcps= FALSE)#plotting r with maxes but no probabilities,
# and not showing the locations of the found changepoints
```

---

poissonProb *Compute predictive probabilities based on Poisson*

Description

Compute the probability of observing the current point, given the current parameters of the poisson for each possible run length. Returns a vector of predictive probabilities from each possible run length, the parameters of the poisson, the most likely lambda of the current poisson, and the current point.

Usage

```
poissonProb(update_params0, update_paramsT, datapt, time, cps, missPts,
  Rlength, skippt = FALSE)
```

Arguments

update_params0	The initialization parameters, corresponding to predicting a changepoint (run length=0)
update_paramsT	The vectors of parameters corresponding to each possible run length, updated with each incoming data point
datapt	the current data point
time	the number of time points passed so far
cps	the current most likely list of changepoints
missPts	the method set to handle missing points
Rlength	the length of the current vector of possible run lengths
skippt	If the current point should be skipped in the updating because it was missing, and missPts was set to skip

Value

Returns a vector of predictive probabilities from each possible run length, the parameters of the gaussian, the most likely mean of the current gaussian, and the current point.

poisson_init *Initialize vectors for poisson probability functions*

Description

Takes in the desired initialization parameters, initializes the vectors needed for the poisson probability function `poisson_update`

Usage

```
poisson_init(init_params = list(a = 1, b = 1), dims)
```

Arguments

`init_params`	The list of parameters to be used for initialization
`dims`	the dimensionality of the data

Value

List of vectors to be used in the iteratively updating algorithm of parameters describing the underlying gaussian distribution of the data.

poisson_update *Update the poisson parameters*

Description

Updates the parameters of the poissons based on each possible run length, after taking into consideration the most recent data point

Usage

```
poisson_update(datapt, update_params0, update_paramsT, Rlength,
  skippt = FALSE)
```

Arguments

`datapt`	the current data point
`update_params0`	The initialization parameters, corresponding to predicting a changepoint (run length=0)
`update_paramsT`	The vectors of parameters corresponding to each possible run length, updated with each incoming data point
`Rlength`	the length of the current vector of possible run lengths
`skippt`	If the current point should be skipped in the updating because it was missing, and missPts was set to skip

Value

The list of the parameters for gaussians corresponding to each possible runlength up to the current data point. Lengths of vectors should correspond the length of the R vector ("run length vector")

`print.ocp` *Print Object*

Description

Print information about the ocpd object.

Usage

```
## S3 method for class 'ocp'
print(x, ...)
```

Arguments

`x`	the object to print
`...`	(optional) additional arguments, ignored.

Examples

```
simdatapts<- c(rnorm(n = 50), rnorm(n=50, 100))
ocpd1<- onlineCPD(simdatapts)
print(ocpd1)
```

`str.ocp` *Object Structure*

Description

Print out information about the ocpd object.

Usage

```
## S3 method for class 'ocp'
str(object, ...)
```

Arguments

`object`	the object to show
`...`	(optional) additional arguments, ignored.

Examples

```
simdatapts<- c(rnorm(n = 50), rnorm(n=50, 100))
ocpd1<- onlineCPD(simdatapts)
str(ocpd1)
```

`studentpdf` *Calculate Student PDF on vector of parameters*

Description

Computes the student pdf from input parameter vectors corresponding to each possible run length for the current time point. Outputs a vector of probabilities for use in the accompanying gaussian functions.

Usage

```
studentpdf(x, mu, var, nu)
```

Arguments

`x`	the current data point
`mu`	vector of means
`var`	var parameter of student pdf, degrees of freedom
`nu`	nu parameter of student pdf (number of points so far)

Value

Vector of student pdf values corresponding to each possible run length, for use in accompanying gaussian probability functions.

`summary.ocp` *Object Summary*

Description

Print out ocpd object summary.

Usage

```
## S3 method for class 'ocp'
summary(object, ...)
```

Arguments

object	the object to summarize
...	(optional) additional arguments, ignored.

Examples

```
simdatapts<- c(rnorm(n = 50), rnorm(n=50, 100))
ocpd1<- onlineCPD(simdatapts)
summary(ocpd1)
```

Index