Package 'essHist'

May 10, 2019

Type Package

Title The Essential Histogram

Version 1.2.2

Date 2019-05-10

Author Housen Li [aut, cre],
Hannes Sieling [aut]

Maintainer Housen Li <housen.li@outlook.com>

Description Provide an optimal histogram, in the sense of probability density estimation and features detection, by means of multiscale variational inference. In other words, the resulting histogram servers as an optimal density estimator, and meanwhile recovers the features, such as increases or modes, with both false positive and false negative controls. Moreover, it provides a parsimonious representation in terms of the number of blocks, which simplifies data interpretation. The only assumption for the method is that data points are independent and identically distributed, so it applies to fairly general situations, including continuous distributions, discrete distributions, and mixtures of both. For details see Li, Munk, Sieling and Walther (2016) <arXiv:1612.07216>.

Depends R (>= 2.15.3)

License GPL-3

LazyData TRUE

Imports Rcpp (>= 0.12.5), graphics, stats, grDevices, utils

LinkingTo Rcpp

Suggests testthat

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-05-10 09:30:03 UTC

R topics documented:

`essHist-package` *The Essential Histogram*

Description

Provide an optimal histogram, in the sense of probability density estimation and features detection, by means of multiscale variational inference. In other words, the resulting histogram servers as an optimal density estimator, and meanwhile recovers the features, such as increases or modes, with both false positive and false negative controls. Moreover, it provides a parsimonious representation in terms of the number of blocks, which simplifies data interpretation. The only assumption for the method is that data points are independent and identically distributed, so it applies to fairly general situations, including continuous distributions, discrete distributions, and mixtures of both. For details see Li, Munk, Sieling and Walther (2016) <arXiv:1612.07216>.

Details

Package:	essHist
Type:	Package
Version:	1.2.2
Date:	2019-05-10
License:	GPL-3

Index:

```
essHistogram    Compute the essential histogram
checkHistogram  Check any estimator by the multiscale confidence set

genIntv         Generate the system of intervals
msQuantile      Simulate the quantile of multiscale statistics

dmixnorm        Compute density function of Gaussian mixtures
pmixnorm        Compute distribution function of Gaussian mixtures
rmixnorm        Generate random number of Gaussian mixtures
paramExample    Output detailed parameters for some famous examples
```

Author(s)

Housen Li [aut, cre], Hannes Sieling [aut]

Maintainer: Housen Li <housen.li@outlook.com>

References

Li, H., Munk, A., Sieling, H., and Walther, G. (2016). The essential histogram. arXiv:1612.07216

Examples

```
# Simulate data
set.seed(123)
type = 'skewed_unimodal'
n = 500
y = rmixnorm(n, type = type)

# Compute the essential histogram
eh = essHistogram(y, plot = FALSE)

# Plot results
#     compute oracle density
x  = sort(y)
od = dmixnorm(x, type = type)
#     compare with orcle density
plot(x, od, type = "l", xlab = NA, ylab = NA, col = "red", main = type)
lines(eh)
legend("topleft", c("Oracle density", "Essential histogram"),
       lty = c(1,1), col = c("red", "black"))



##### Evaluate other method
set.seed(123)
# Data: mixture of Gaussians "harp"
n = 500
y = rmixnorm(n, type = 'harp')

# Oracle density
x = sort(y)
ho = dmixnorm(x, type = 'harp')

# R default histogram
h  = hist(y, plot = FALSE)

# Check R default histogram to local multiscale constraints
b = checkHistogram(h, y, ylim=c(-0.1,0.16))
lines(x, ho, col = "red")
rug(x, col = 'blue')
legend("topright", c("R-Histogram", "Truth"), col = c("black", "red"), lty = c(1,1))
```

`checkHistogram` *Check any histogram estimator by means of the multiscale confidence set*

Description

Provide the locations, i.e., intervals, where features are potentially missing (a.k.a. false negatives), and the break-points that are potentially redundant (a.k.a. false positives), by means of the multiscale confidence set.

Usage

```
checkHistogram(h, x, alpha = 0.1, q = NULL, intv = NULL,
               mode = ifelse(anyDuplicated(x),"Gen","Con"),
               plot = TRUE, xlim = NULL, ylim = NULL,
               xlab = "", ylab = "", yaxt = "n", ...)
```

Arguments

h	a numeric vector specifying values of a histogram at sample points; or a `hitogram` class object (i.e. the return value of `hist`).
x	a numeric vector containing the data.
alpha	significance level, default as 0.1, see also `essHistogram`.
q	threshold of the multiscale constraint; by default, `q` is chosen as the (1-`alpha`)-quantile of the null distribution of the multiscale statistic via Monte Carlo simulation, see also `msQuantile`.
intv	a data frame provides the system of intervals on which the multiscale statistic is defined. The data frame constains the following two columns `left` left index of an interval `right` right index of an interval By default, it is set to the sparse interval system proposed by Rivera and Walther (2013), see also Li et al. (2016).
mode	`"Con"` for continuous distribution functions `"Gen"` for general (possibly with discontinuous) distribution functions By default, `"Con"` is chosen if there is no tied observations; otherwise, `"Gen"` is chosen; see Li et al. (2016) for further details.
plot	logical. If `TRUE`, the input estimator is potted, together with evaluation information. More precisely, at the very bottom, intervals where local constaints are violated are plotted. In the middle short vertical lines that indicate possibly removable change-points are drawn above a light blue horizontal line. Right below the light blue line, it plots a horizontal gray scale strap, the darkness of which reflects the number of violation intervals covering a given location, as a summary of violation information.
xlim, ylim	numeric vectors of length 2 (default `xlim = range(y), ylim = NULL`): see `plot`.
xlab	a title for the `x` axis (default empty string): see `title` and `plot`.
ylab	a title for the `y` axis (default empty string): see `title` and `plot`.
yaxt	A character which specifies the `y` axis type (default `"n"`): see `par`.
...	further arguments and `graphical parameters` passed to `plot` (if `plot` = `TRUE`).

Details

This function presents a visualization: the upper part plots the given histogram; in the middle part short vertical lines mark all removable break-points; in the lower part intervals of violation are shown, and a graybar below the middle horizontal line (blue) sumarizes such violations with the darkness scaling with the number of violation intervals covering a location. See Examples below and Li et al. (2016) for further details.

Value

A list consists of one data frame, and one numeric vector:

`violatedIntervals`
: A data frame provides the intervals where the corresponding local side constraint is violated; an empty data frame if there is no violation. It constains the following four columns

  `leftIndex` left index of an interval

  `rightIndex` right index of an interval

  `leftEnd` left end point of an interval

  `rightEnd` right end point of an interval

  An empty `data.frame` is returned if there is no violation.

`removableBreakpoints`
: A numeric vector contains all removable breakpoints, with zero length if there is no removable breakpoint.

Note

The argument `intv` is internally adjusted ensure it contains no empty intervals in case of tied observations. Only the intervals on which the input histogram is constant will be checked! All the printing messages can be disabled by calling `suppressMessages`.

References

Li, H., Munk, A., Sieling, H., and Walther, G. (2016). The essential histogram. arXiv:1612.07216.

See Also

`essHistogram`, `genIntv`, `msQuantile`

Examples

```
set.seed(123)
# Data: mixture of Gaussians "harp"
n = 500
y = rmixnorm(n, type = 'harp')

# Oracle density
x = sort(y)
ho = dmixnorm(x, type = 'harp')
```

```
# R default histogram
h  = hist(y, plot = FALSE)

# Check R default histogram to local multiscale constriants
b = checkHistogram(h, y, ylim=c(-0.1,0.16))
lines(x, ho, col = "red")
rug(x, col = 'blue')
legend("topright", c("R-Histogram", "Truth"), col = c("black", "red"), lty = c(1,1))
```

Essential Histogram *The Essential Histogram*

Description

Compute the essential histogram via (pruned) dynamic programming.

Usage

```
essHistogram(x, alpha = 0.5, q = NULL, intv = NULL, plot = TRUE,
             mode = ifelse(anyDuplicated(x),"Gen","Con"),
             xname = deparse(substitute(x)), ...)
```

Arguments

x	a numeric vector containing the data.
alpha	significance level; default as 0.5. One should set `alpha = 0.1` or even smaller if confidence statements have to be made, while one can set `alpha = 0.9` if the goal is to explore the data for potential features with tolerance to false positives. The default value is only a trade-off.
q	threshold value; by default, q is chosen as the (1-alpha)-quantile of the null distribution of the multiscale statistic via Monte Carlo simulation, see also `msQuantile`.
intv	a data frame provides the system of intervals on which the multiscale statistic is defined. The data frame constains the following two columns `left` left index of an interval `right` right index of an interval By default, it is set to the sparse interval system proposed by Rivera and Walther (2013), see also Li et al. (2016).
plot	logical. If `TRUE` (default), a histogram is plotted, otherwise a list of breaks and counts is returned. In the latter case, a warning is used if (typically graphical) arguments are specified that only apply to the `plot = TRUE` case.
mode	`"Con"` for continuous distribution functions `"Gen"` for general (possibly with discontinuous) distribution functions By default, `"Con"` is chosen if there is no tied observations; otherwise, `"Gen"` is chosen; see Li et al. (2016) for further details.
xname	a character string with the actual x argument name.
...	further arguments and `graphical parameters` passed to `plot.histogram` and thence to `title` and `axis` (if `plot = TRUE`).

Details

The essential histogram is defined as the histogram with least blocks within the multiscale constraint. The one with highest likelihood is picked if there are more than one solutions. The essential histogram involves only one parameter `q`, the threshold of the multiscale constraint. Such a parameter can be chosen by means of the significance level `alpha`, which leads to nature statistical significance statements for the multiscale constraint. The computational complexity is often linear in terms of sample size, although the worst complexity bound is quadratic up to a log-factor in case of the sparse interval system. See Li et al. (2016) for further details.

Value

An object of class "`histogram`", which is of the same class as returned by function `hist`.

Note

The argument `intv` is internally adjusted to ensure it contains no empty intervals, especially in case of tied observations. The first block of the returned histogram is a closed interval, and the rest blocks are left open right closed intervals. All the printing messages can be disabled by calling `suppressMessages`.

References

Li, H., Munk, A., Sieling, H., and Walther, G. (2016). The essential histogram. arXiv:1612.07216.

Rivera, C., & Walther, G. (2013). Optimal detection of a jump in the intensity of a Poisson process or in a density with likelihood ratio statistics. Scand. J. Stat. 40, 752–769.

See Also

`checkHistogram`, `genIntv`, `hist`, `msQuantile`

Examples

```
# Simulate data
set.seed(123)
type = 'skewed_unimodal'
n = 500
y = rmixnorm(n, type = type)

# Compute the essential histogram
eh = essHistogram(y, plot = FALSE)

# Plot results
#     compute oracle density
x  = sort(y)
od = dmixnorm(x, type = type)
#     compare with orcle density
plot(x, od, type = "l", xlab = NA, ylab = NA, col = "red", main = type)
lines(eh)
legend("topleft", c("Oracle density", "Essential histogram"),
       lty = c(1,1), col = c("red", "black"))
```

Generate Intervals *Generate the system of intervals*

Description

Generate the system of intervals on which the multiscale statistic is defined, see Li et al. (2016).

Usage

```
genIntv(n, type = c("Sparse", "Full"))
```

Arguments

n	number of observations.
type	type of interval system. `type = "Sparse"` (default) is the sparse system proposed by Rivera and Walther (2013), see also Li et al. (2016). `type = "Full"` is the system of all possible intervals with end-index ranging from 1 to n.

Value

A data frame provides the system of intervals, and consists two columns

left	left index of an interval
right	right index of an interval

References

Li, H., Munk, A., Sieling, H., and Walther, G. (2016). The essential histogram. arXiv:1612.07216.

Rivera, C., & Walther, G. (2013). Optimal detection of a jump in the intensity of a Poisson process or in a density with likelihood ratio statistics. Scand. J. Stat. 40, 752–769.

See Also

checkHistogram, essHistogram, msQuantile

Examples

```
n    = 5
intv = genIntv(n,"Full")
print(intv)
```

`Mixed normals` *The mixture of normal distributions*

Description

Density, distribution function and random generation for the mixture of normals with each component specified by `mean` and `sd`, and mixture weights by `prob`. `paramExample` gives detailed parameters for some examples specified by `type`.

Usage

```
dmixnorm(x, mean = c(0), sd = rep(1,length(mean)),
          prob = rep(1,length(mean)), type = NULL, ...)
pmixnorm(x, mean = c(0), sd = rep(1,length(mean)),
          prob = rep(1,length(mean)), type = NULL, ...)
rmixnorm(n, mean = c(0), sd = rep(1,length(mean)),
          prob = rep(1,length(mean)), type = NULL)
paramExample(type)
```

Arguments

`x`	vector of locations.
`n`	integer; number of observations.
`mean`	vector of means for each mixture component.
`sd`	vector of standard deviations for each mixture component. Default is of unit variance for each component.
`prob`	vector of prior probability for each mixture component (i.e. mixture weights). All nonnegative values are allowed, and automatically recaled to ensure their sum equal to 1. Default is of equal probability for each component.
`type`	a (case insensitive) character string of example name; It includes examples from Marron & Wand (1992): "MW1", ..., "MW15", or equivalently "gauss", "skewed_unimodal", "strong_skewed", "kurtotic_unimodal", "outlier", "bimodal", "separated_bimodal", "skewed_bimodal", "trimodal", "claw", "double_claw", "asymmetric_claw", "asymmetric_double_claw", "smooth_comb", "discrete_comb"; It also includes "harp" example from Li et al. (2016).
`...`	further arguments passed to `dnorm` and `pnorm`.

Details

Users either provide, optionally, `mean`, `sd` and `prob`; or `type`. In case of providing `type`, the values of `mean`, `sd` and `prob` are ignored.

The default case is standard normal, the same as `dnorm`, `pnorm` and `rnorm`.

Value

`dmixnorm` gives the density, `pmixnorm` gives the distribution function, and `rmixnorm` generates random deviates.

The length of the result is determined by `n` for `rmixnorm`, and is the length of `x` for `dmixnorm` and `pmixnorm`.

`paramExample` gives a data frame with components `mean`, `sd` and `prob`.

References

Li, H., Munk, A., Sieling, H., and Walther, G. (2016). The essential histogram. arXiv:1612.07216.

Marron, J. S., & Wand, M. P. (1992). Exact mean integrated squred error. Ann. Statist., 20(2), 712–736.

See Also

Normal for standard normal distributions; Distributions for other standard distributions.

Examples

```
## Example harp
type = "harp"
#     generate random numbers
n = 500
Y = rmixnorm(n, type = type)
#     compute the density
x = seq(min(Y), max(Y), length.out = n)
f = dmixnorm(x, type = type)
#     compute the distribution
F = pmixnorm(x, type = type)
#     plots
op = par(mfrow = c(1,2))
plot(x, f, type = "l", main = "Harp Density")
rug(Y, col = 'red')
plot(x, F, type = "l", main = "Harp Distribution")
rug(Y, col = 'red')
par(op)
```

`Multiscale Quantiles` *Quantile of the multiscale statistics*

Description

Simulate quantiles of the multiscale statistics under any continuous distribution function.

Usage

```
msQuantile(n, alpha = c(0.5), nsim = 5e3, is.sim = (n < 1e4),
           intv = genIntv(n), mode = c("Con", "Gen"), ...)
```

Arguments

`n`	number of observations.
`alpha`	significance level; default as 0.5, see also `essHistogram`. Like `quantile`, it can also be a vector.
`nsim`	numer of Monte Carlo simulations.
`is.sim`	logical. If `TRUE` (default if `n` < 10,000) the quantile is determined via Monte Carlo simulations, which might take a long time; otherwise (default if `n` >= 10,000) it uses the quantile with `n` = 10,000, which has been precomputed and stored.
`intv`	a data frame provides the system of intervals on which the multiscale statistic is defined. The data frame constains the following two columns `left` left index of an interval `right` right index of an interval By default, it is set to the sparse interval system proposed by Rivera and Walther (2013), see `genIntv` and also Li et al. (2016).
`mode`	`"Con"` for continuous distribution functions (default) `"Gen"` for general (possibly with discontinuous) distribution functions See Li et al. (2016) for further details.
`...`	further arguments passed to function `quantile`.

Details

Empirically, it turns out that the quantile of the multiscale statistic converges fast to that of the limit distribution as the number of observations `n` increases. Thus, for the sake of computational efficiency, the quantile with `n` = 10,000 are used by default for that with `n` > 10,000, which has already been precomputed and stored. Of course, for arbitrary sample size `n`, one can always simulate the quantile by setting `is.sim = TRUE`, and use the precomputed value by setting `is.sim = FALSE`. For a given sample size `n`, simulations are once computed, and then automatically recorded in the R memory for later usage. For memory efficiency, only the last simulation is stored.

Value

A vector of length `length(alpha)` is returned, the same structure as returned by funtion `quantile` with option `names = FALSE`; The values are the (1-`alpha`)-quantile(s) of the null distribution of the multiscale statistic via Monte Carlo simulation, corresponding to (1-`alpha`)-confidence level(s). See Li et al. (2016) for further details.

Note

All the printing messages can be disabled by calling `suppressMessages`.

References

Li, H., Munk, A., Sieling, H., and Walther, G. (2016). The essential histogram. arXiv:1612.07216.

Rivera, C., & Walther, G. (2013). Optimal detection of a jump in the intensity of a Poisson process or in a density with likelihood ratio statistics. Scand. J. Stat. 40, 752–769.

See Also

checkHistogram, essHistogram, genIntv

Examples

```
n    = 100 # number of observations
nsim = 100 # number of simulations

alpha = c(0.1, 0.9) # significance level
q     = msQuantile(n, alpha, nsim)

print(q)
```

Index