Package 'LeArEst'

April 8, 2019

Type Package

Title Border and Area Estimation of Data Measured with Additive Error

Version 1.0.0

Date 2019-04-05

Maintainer Petar Taler <petar@mathos.hr>

Description Provides methods for estimating borders of uniform distribution on
the interval (one-dimensional) and on the elliptical domain (two-dimensional)
under measurement errors. For one-dimensional case, it also estimates the
length of underlying uniform domain and tests the hypothesized length against
two-sided or one-sided alternatives. For two-dimensional case, it estimates
the area of underlying uniform domain. It works with numerical inputs as well
as with pictures in JPG format.

License GPL-2

LazyData TRUE

RoxygenNote 6.1.1

Imports conicfit, doParallel, foreach, graphics, jpeg, opencpu (>=
2.0.0), parallel, stats, utils

NeedsCompilation no

Encoding UTF-8

Author Mirta Bensic [aut],
Safet Hamedovic [aut],
Kristian Sabo [aut],
Petar Taler [aut, cre]

Repository CRAN

Date/Publication 2019-04-08 09:42:45 UTC

R topics documented:

---

areaest   *Performs area estimation of the numerically described object in plane.*

Description

Use this function if you have a data set of uniformly distributed points on an elliptical domain in
the plane but captured with additive errors. The estimation algorithm takes many horizontal and
vertical, or star-shaped slices of the object. Length estimation procedure is conducted on each slice
and in that way the set of edge points is obtained. An ellipse or a circle is fitted to these edge
points by function EllipseDirectFit or CircleFitByPratt from the package conicfit and its
semi-axes and area are returned as a result. Function optionally plots input points, calculated edge
points and the resulting ellipse or circle.

Usage

```
areaest(data, nrSlices = 10, error = c("laplace", "gauss", "student"),
  var.est = c("MM", "ML"), var = NULL, plot = FALSE,
  parallel = FALSE, slicing = c("hv", "star"),
  representation = c("ellipse", "circle"))
```

Arguments

data	Two-column data matrix containing the points that describe observed object. First column represents *x* coordinate of the point, while second column represents *y* coordinate.
nrSlices	Number of slices applied for plain data cutting. Defaults to 10.
error	A character string specifying the error distribution. Must be one of "laplace", "gauss" or "student". Can be abbreviated.
var.est	A character string specifying the method of error variance estimation. Must be given if var is not given. Can be "MM" (Method of Moments) or "ML" (Maximum Likelihood).
var	Explicit error variance. Needs to be given if var.est is not given.
plot	Logical parameter (TRUE or FALSE) that determines whether to plot given object, calculated edge points and the resulting ellipse. Defaults to FALSE.
parallel	Logical parameter (TRUE or FALSE) that determines whether to perform estimation procedure in a parallel manner. Can shorten estimation time if many border points need to be calculated. Defaults to FALSE.
slicing	A character string specifying the method of slicing. Can be "hv" (horizontal and vertical slicing) or "star" (star-shaped slicing). Can be abbreviated.
representation	A character string specifying the shape of an observed object. Can be "ellipse" or "circle". Can be abbreviated.

Value

List containing:

- area: Estimated area of the object,
- points: Set of calculated object's edge points,
- semiaxes: Resulting ellipse's semi-axes or circle radius.

Examples

```
# load a data set representing the ellipse with additive Gaussian error,
# run area estimation on it, and plot the results
inputfile <- system.file("extdata", "ellipse_3_4_0.1_gauss.txt", package = "LeArEst")
inputdata <- read.table(inputfile)
area <- areaest(inputdata, error = "gauss", var.est = "ML", plot = TRUE,
                slicing = "hv", representation = "ellipse")

# load a data set representing the ellipse with additive Laplacian error,
# run area estimation on it, and plot the results
inputfile <- system.file("extdata", "ellipse_3_4_0.1_laplace.txt", package = "LeArEst")
inputdata <- read.table(inputfile)
area <- areaest(inputdata, error = "laplace", var = 0.1, nrSlices = 5, plot = TRUE,
                slicing = "star", representation = "ellipse")
```

---

lengthest   *Performs width estimation for a numerical data set.*

Description

Function `lengthest()` computes the length of an interval which is the domain of uniform distribution from data contaminated with additive error. The additive error can be chosen as Laplace, Gauss or scaled Student distribution with 1 - 5 degrees of freedom.

Usage

```
lengthest(x, error = c("laplace", "gauss", "t1", "t2", "t3", "t4", "t5"),
  sd = NULL, sd.est = c("MM", "ML"), conf.level = 0.95)
```

Arguments

x	Vector of input data.
error	A character string specifying the error distribution. Must be one of "laplace", "gauss", "t1", "t2", "t3", "t4", "t5". Can be abbreviated.
sd	Explicit error standard deviation. Needs to be given if sd.est is not given.
sd.est	A character string specifying the method of error standard deviation estimation. Must be given if sd is not given. Can be "MM" (Method of Moments) or "ML" (Maximum Likelihood).
conf.level	Confidence level of the confidence interval.

Value

List containing:

- error.type: A character string describing the type of the error distribution.
- radius: Estimated half-width of uniform distribution.
- sd.error: Error standard deviation, estimated or given.
- conf.level: Confidence level of the confidence interval.
- method: A character string indicating what method for computing a confidence interval was used (asymptotic distribution of ML or likelihood ratio statistic).
- conf.int: The confidence interval for half-width.

Examples

```
# generate uniform data with additive error and run a length estimation on it
sample_1 <- runif(1000, -1, 1)
sample_2 <- rnorm(1000, 0, 0.1)
sample <- sample_1 + sample_2
out <- lengthest(x = sample, error = "gauss", sd.est = "MM", conf.level = 0.90)
```

---

lengthtest   *Test for uniform distribution width.*

Description

Function `lengthtest()` tests the hypothesized uniform domain width against two-sided or one-sided alternatives from data contaminated with additive error. The additive error can be chosen as Laplace, Gauss or scaled Student distribution with 1 - 5 degrees of freedom.

Usage

```
lengthtest(x, error = c("laplace", "gauss", "t1", "t2", "t3", "t4",
  "t5"), alternative = c("two.sided", "greater", "less"), sd = NULL,
  null.a = NULL, sd.est = c("MM", "ML"), conf.level = 0.95)
```

Arguments

x	Vector of input data
error	A character string specifying the error distribution. Must be one of "laplace", "gauss", "t1", "t2", "t3", "t4", "t5". Can be abbreviated.
alternative	A character string specifying the alternative hypothesis, must be one of "two.sided", "greater" or "less". Can be abbreviated.
sd	Explicit error standard deviation. Needs to be given if var.sd is not given.
null.a	Specified null value being tested.

sd.est	A character string specifying the method of error standard deviation estimation. Must be given if `sd` is not given. Can be "MM" (Method of Moments) or "ML" (Maximum Likelihood).
conf.level	Confidence level of the confidence interval.

Value

List containing:

- error.type: A character string describing the type of the error distribution,
- radius: Estimated half-width of uniform distribution,
- sd.error: Error standard deviation, estimated or given,
- conf.level: Confidence level of the confidence interval,
- alternative: A character string describing the alternative hypothesis,
- method: A character string indicating what method for testing was used (asymptotic distribution of MLE or likelihood ratio statistic),
- conf.int: The confidence interval for half-width,
- null.a: null value being tested,
- p.value: p-value of the test,
- tstat: the value of the test statistic.

Examples

```
# generate uniform data with additive error and run a hypothesis testing on it
sample_1 <- runif(1000, -1, 1)
sample_2 <- rnorm(1000, 0, 0.1)
sample <- sample_1 + sample_2
out <- lengthtest(x = sample, error = "gauss", alternative = "greater",
                  sd.est = "MM", null.a = 0.997, conf.level = 0.95)
```

---

startweb.area	*Opens default web browser and loads a web page for area estimation.*

Description

Function `startweb.area()` opens a web browser and loads the page for area estimation of object shown in a picture. Use this function if you can isolate a part of the picture with uniformly distributed dots on an elliptical domain with unclear borders. Sequence of actions on the web page is as follows:

1. Load a picture in JPG format
2. Click on upper left and lower right corner of a rectangle surrounding observed object so the rectangle is drawn
3. Set data and estimation parameters
4. Click on Estimate

Usage

```
startweb.area()
```

Details

The area estimation algorithm takes many horizontal and vertical (if "horizontal + vertical" slicing is selected) or star-shaped (if "star" slicing is selected) slices of the object. Length estimation procedure is conducted on each slice and in that way set of edge points is obtained. Lastly, ellipse or circle is fitted on that set of points by function `EllipseDirectFit` or `CircleFitByPratt` from the package `conicfit` and area of that ellipse or circle is returned as the result. The area is measured in pixels, as well as percentage of the whole image.

Parameters that can be set on the web page are as follows:

Data parameters

Levels of grey Number of colors (shades of grey) used in analysis.

Box size The algorithm takes each pixel of a picture and maps it to box_size * box_size matrix. It is done in a way that the brightness of the observed pixel dictates the quantity of dots in mentioned matrix. Distribution of dots in matrix is uniform. Ultimately, length estimation is done on the set of the resulting matrices.

Line thickness Width of the slice, i.e. the maximum length between surrounding pixel and the drawn line so that pixel is to be taken into account for length estimation. All surrounding pixels are orthogonally projected on the central line.

Number of slices Number of slices after cutting in one direction. Defaults to 10. Slices are equally thick in both directions. Smaller number of cuts will be automatically applied for smaller dimension if the chosen rectangle is not a square.

Slicing Sets slicing method for the edge point estimation. Can be "horizontal + vertical" or "star".

Parallelization Sets whether to distribute area estimation on multiple CPU cores. If set to On, total number of cores - 1 are used.

Object brightness Sets whether observed object is bright or dark.

Represent object as Represent estimated object as an ellipse or as a circle.

Estimation parameters

Error distribution Type of the error distribution. Can be Gauss, Laplace, T1, T2, T3, T4 or T5 (Student).

Error standard deviation Estimation method for the error standard deviation. Can be Maximum Likelihood (ML) or the Method of Moments. If one does not want to estimate the deviation but to explicitly enter it, he should choose "Enter value" and enter the deviation in the lower field.

Note

In order to have quadratic pixels on the screen, please use proportional screen resolution. In the case of modern LCD (LED) displays, these are usually native screen resolutions. If your display has aspect ratio width:height = 16:9, these resolutions are 1280x720, 1600x900, 1920x1080, etc. In the case od 16:10 display, use 1280x800, 1440x900, 1920x1200, etc. If you use nonproportional screen resolution, pixels on the screen will not be quadratic, so estimated values measured in pixels may not be correct.

Examples

```
# open the web page for area estimation of an object shown in the picture
startweb.area()
```

---

startweb.esttest	*Opens default web browser and loads a web page for length estimation and testing.*

Description

Function `startweb.esttest()` opens a web browser and loads the page for length estimation and hypothesis testing of data read from a picture. Use this function if you would like to compute the length of an interval, which is the domain of uniform distribution, but the data are contaminated with additive error. Sequence of actions on the web page is as follows:

1. Load a picture in JPG format
2. Click on start and end point of the line passing through the observed object
3. Set data preparation parameters
4. Click on generate data so the data are prepared
5. • set estimation parameters (see `lengthest`) and click on Estimate, or
   • set testing parameters (see `lengthtest`) and click on Test.

Usage

```
startweb.esttest()
```

Details

Parameters that can be set on the web page are as follows:

Data parameters

Levels of grey Number of colors (shades of grey) used in analysis.

Box size The algorithm takes each pixel of a picture and maps it to box_size * box_size matrix. It is done in a way that the brightness of the observed pixel dictates the quantity of *dots* in mentioned matrix. Distribution of dots in matrix is uniform. Ultimately, length estimation or testing is done on the set of the resulting matrices.

Line thickness Width of the line, i.e. the maximum length between surrounding pixel and the drawn line so that pixel is to be taken into account for length estimation or hypothesis testing. All surrounding pixels are orthogonally projected on the line.

Observed object is... Sets whether observed object is bright or dark.

Estimation parameters

Error distribution The type of the error distribution. Can be Gauss, Laplace, T1, T2, T3, T4 or T5 (Student).

Error standard deviation Estimation method for the error standard deviation. Can be Maximum Likelihood (ML) or the Method of Moments. If one does not want to estimate the deviation but to explicitly enter it, he should choose "Enter value" and enter the deviation in the lower field.

Confidence level Confidence level of the confidence interval.

Testing parameters

$latex$ Specified null value being tested (measured in pixel width or percentage of image width).

Alternative The alternative hypothesis. Can be less, greater or two-sided.

Results

- Length expressed in pixel width, as well as in percentage of the whole image's width
- Standard deviation
- Method - Asymptotic distribution of MLE or LR statistic
- Confidence interval
- p-value of the test and the value of the test statistic (if hypothesis testing has been performed)

Note

In order to have quadratic pixels on the screen, please use proportional screen resolution. In the case of modern LCD (LED) displays, these are usually native screen resolutions. If your display has aspect ratio width:height = 16:9, these resolutions are 1280x720, 1600x900, 1920x1080, etc. In the case od 16:10 display, use 1280x800, 1440x900, 1920x1200, etc. If you use nonproportional screen resolution, pixels on the screen will not be quadratic, so estimated values measured in pixels may not be correct.

Examples

```
# open the web page for length estimation and hypothesis testing of an
# object shown in the picture
startweb.esttest()
```

Index