roahd (version 1.4.3)

outliergram: Outliergram for univariate functional data sets

Description

This function performs the outliergram of a univariate functional data set, possibly with an adjustment of the true positive rate of outliers discovered under assumption of gaussianity.

Usage

```
outliergram(
  fData,
  MBD_data = NULL,
  MEI_data = NULL,
  p_check = 0.05,
  Fvalue = 1.5,
  adjust = FALSE,
  display = TRUE,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  ...
)
```

Arguments

`fData`	the univariate functional dataset whose outliergram has to be determined.
`MBD_data`	a vector containing the MBD for each element of the dataset. If missing, MBDs are computed.
`MEI_data`	a vector containing the MEI for each element of the dataset. If not not provided, MEIs are computed.
`p_check`	percentage of observations with either low or high MEI to be checked for outliers in the secondary step (shift towards the center of the dataset).
`Fvalue`	the F value to be used in the procedure that finds the shape outliers by looking at the lower parabolic limit in the outliergram. Default is `` `1.5` ``. You can also leave the default value and, by providing the parameter `` `adjust` ``, specify that you want `` `Fvalue` `` to be adjusted for the dataset provided in `` `fData` ``.

adjust either `FALSE` if you would like the default value for the inflation factor, $F = 1.5$, to be used, or a list specifying the parameters required by the adjustment.

"`N_trials`": the number of repetitions of the adjustment procedure based on the simulation of a gaussian population of functional data, each one producing an adjusted value of F, which will lead to the averaged adjusted value $\bar{F}$. Default is 20;

"`trial_size`": the number of elements in the gaussian population of functional data that will be simulated at each repetition of the adjustment procedure. Default is `5 * fData$N`;

"`TPR`": the True Positive Rate of outliers, i.e. the proportion of observations in a dataset without shape outliers that have to be considered outliers. Default is `2 * pnorm( 4 * qnorm( 0.25 ) )`;

"`F_min`": the minimum value of F, defining the left boundary for the optimization problem aimed at finding, for a given dataset of simulated gaussian data associated to `fData`, the optimal value of F. Default is 0.5;

"`F_max`": the maximum value of F, defining the right boundary for the optimization problem aimed at finding, for a given dataset of simulated gaussian data associated to `fData`, the optimal value of F. Default is 20;

"`tol`": the tolerance to be used in the optimization problem aimed at finding, for a given dataset of simulated gaussian data associated to `fData`, the optimal value of F. Default is `1e-3`;

"`maxiter`": the maximum number of iterations to solve the optimization problem aimed at finding, for a given dataset of simulated gaussian data associated to `fData`, the optimal value of F. Default is `100`;

"`VERBOSE`": a parameter controlling the verbosity of the adjustment process;

display either a logical value indicating whether you want the outliergram to be displayed, or the number of the graphical device where you want the outliergram to be displayed.

xlab a list of two labels to use on the x axis when displaying the functional dataset and the outliergram

ylab a list of two labels to use on the y axis when displaying the functional dataset and the outliergram;

main a list of two titles to be used on the plot of the functional dataset and the outliergram;

... additional graphical parameters to be used *only* in the plot of the functional dataset

Value

Even when used graphically to plot the outliergram, the function returns a list containing:

`Fvalue`: the value of the parameter F used;

`d`: the vector of values of the parameter d for each observation (distance to the parabolic border of the outliergram);

`ID_outliers`: the vector of observations id corresponding to outliers.

Adjustment

When the adjustment option is selected, the value of F is optimized for the univariate functional dataset provided with `fData`. In practice, a number `adjust$N_trials` of times a synthetic population (of size `adjust$trial_size` with the same covariance (robustly estimated from data) and centerline as `fData` is simulated without outliers and each time an optimized value F_i is computed so that a given proportion (`adjust$TPR`) of observations is flagged as outliers. The final value of `F` for the outliergram is determined as an average of $F_1, F_2, \ldots, F_{N_{trials}}$. At each time step the optimization problem is solved using `stats::uniroot` (Brent's method).

References

Arribas-Gil, A., and Romo, J. (2014). Shape outlier detection and visualization for functional data: the outliergram, *Biostatistics*, 15(4), 603-619.

See Also

`fData`, `MEI`, `MBD`, `fbplot`

Examples

Run this code

```
# NOT RUN {
set.seed(1618)

N <- 200
P <- 200
N_extra <- 4

grid <- seq(0, 1, length.out = P)

Cov <- exp_cov_function(grid, alpha = 0.2, beta = 0.8)

Data <- generate_gauss_fdata(
  N = N,
  centerline = sin(4 * pi * grid),
  Cov = Cov
)

Data_extra <- array(0, dim = c(N_extra, P))

Data_extra[1, ] <- generate_gauss_fdata(
  N = 1,
  centerline = sin(4 * pi * grid + pi / 2),
  Cov = Cov
)

Data_extra[2, ] <- generate_gauss_fdata(
  N = 1,
  centerline = sin(4 * pi * grid - pi / 2),
  Cov = Cov
)

Data_extra[3, ] <- generate_gauss_fdata(
  N = 1,
  centerline = sin(4 * pi * grid + pi / 3),
  Cov = Cov
)

Data_extra[4, ] <- generate_gauss_fdata(
  N = 1,
  centerline = sin(4 * pi * grid - pi / 3),
  Cov = Cov
)
```

```
Data <- rbind(Data, Data_extra)

fD <- fData(grid, Data)

# Outliergram with default Fvalue = 1.5
outliergram(fD, display = TRUE)

# Outliergram with Fvalue enforced to 2.5
outliergram(fD, Fvalue = 2.5, display = TRUE)

# }
# NOT RUN {
# Outliergram with estimated Fvalue to ensure TPR of 1%
outliergram(
  fData = fD,
  adjust = list(
    N_trials = 10,
    trial_size = 5 * nrow(Data),
    TPR = 0.01,
    VERBOSE = FALSE
  ),
  display = TRUE
)
# }
# NOT RUN {
# }
```

Run the code above in your browser using DataLab