Package ‘FunctionalCalibration’

July 21, 2025

Type Package

Title Aggregated Functional Data Calibration using Splines and Wavelets

Version 1.0.0

Description Implements methods for calibrating an aggregated functional data model using wavelets or splines. Each aggregated curve is modeled as a linear combination of component functions and known weights. The component functions are estimated using wavelets or splines. The package is based on dos Santos Sousa (2024) <doi:10.1515/mcma-2023-2016>
and Saraiva and Dias (2009) <doi:10.47749/T/UNICAMP.2009.471073>.

URL `https://github.com/VitorRibasP/FunctionalCalibration`

Imports wavethresh

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 3.5)

NeedsCompilation no

Author Vitor Perrone [aut, cre] (ORCID:
<`https://orcid.org/0009-0009-6923-7712`>),
Alex Sousa [aut] (ORCID: <`https://orcid.org/0000-0001-5887-3638`>)

Maintainer Vitor Perrone <`vitor.perrone10@gmail.com`>

Repository CRAN

Date/Publication 2025-06-19 07:30:05 UTC

Contents

functional_calibration_splines 2
functional_calibration_wavelets 3
plot_aggregated_curve 4
simulated_data 5
weight_estimation 6

Index 8

`functional_calibration_splines` *Functional Data Calibration with Splines*

Description

This function performs functional calibration based on the following model:

$$A_i(x_m) = \sum_{l=1}^{L} y_{il}\alpha_l(x_m) + e_i(x_m), \quad i = 1, ..., I, \quad m = 1, ..., M = 2^J$$

where the functions $\alpha_l(x)$ are estimated using spline basis functions.

In matrix notation, the model is represented as:

$$A = \alpha y + e$$

Usage

```
functional_calibration_splines(data, weights, x, n_functions = 10)
```

Arguments

`data` A matrix M x I where each column represents one sample of the aggregated function — the matrix A in the model.

`weights` A matrix L x I representing the weight values associated with each sample — the matrix y in the model.

`x` A numeric vector of values at which the function is evaluated.

`n_functions` Number of spline basis functions to be used for estimating $\alpha_l(x)$.

Value

The function returns a list containing two objects.

`alpha` A matrix with the estimated functional coefficients α.

`Plots` A list of plot objects, each representing the corresponding function $\alpha_l(x)$.

References

Saraiva, M. A., & Dias, R. (2009). Analise não-parametrica de dados funcionais: uma aplicação a quimiometria (Doctoral dissertation, Master's thesis, Universidade Estadual de Campinas, Campinas).

Examples

```
functional_calibration_splines(simulated_data$data, simulated_data$weights, simulated_data$x)
functional_calibration_splines(simulated_data$data, simulated_data$weights, simulated_data$x, 12)
```

functional_calibration_wavelets

Functional Data Calibration with Wavelets

Description

This function performs functional calibration based on the following model:

$$A_i(x_m) = \sum_{l=1}^{L} y_{il}\alpha_l(x_m) + e_i(x_m), \quad i = 1, ..., I, \quad m = 1, ..., M = 2^J$$

where the functions $\alpha_l(x)$ are estimated using wavelet decomposition.

In matrix notation, the model is represented as:

$$A = \alpha y + e$$

Usage

```
functional_calibration_wavelets(
  data,
  weights,
  wavelet = "DaubExPhase",
  method = "bayesian",
  tau = 1,
  p = NULL,
  sigma = NULL,
  MC = FALSE,
  type = "soft",
  singular = FALSE,
  x = NULL
)
```

Arguments

`data`	A matrix M x I where each column represents one sample of the aggregated function — the matrix A in the model.
`weights`	A matrix L x I representing the weight values associated with each sample — the matrix y in the model.
`wavelet`	A string indicating the wavelet family to be used in the Discrete Wavelet Transform (DWT).
`method`	A string specifying the shrinkage method applied to the empirical wavelet coefficients. Options are: "bayesian", "universal", "sure", "probability", or "cv".
`tau`	A numeric value for the τ parameter in the Bayesian shrinkage. If `NULL`, it is estimated from the data.

p	A numeric value for the p parameter in the Bayesian shrinkage. If NULL, it is estimated from the data.
sigma	A numeric value for the σ parameter in the Bayesian shrinkage. If NULL, it is estimated from the data.
MC	A logical evaluating to TRUE or FALSE indicating if the integrals in the Bayesian shrinkage are approximated using Monte Carlo simulation.
type	A string indicating whether the thresholding should be "soft" or "hard" (applies only when the method is not "bayesian").
singular	A logical evaluating to TRUE or FALSE indicating if it adds a small constant (1e-10) to the diagonal of yy^T to stabilize the matrix inversion.
x	A numeric vector of values at which the function is evaluated. If NULL, the default is the sequence 1:nrow(data).

Value

The function returns a list containing two objects:

alpha A matrix with the estimated functional coefficients α.

Plots A list of plot objects, each representing the corresponding function $\alpha_l(x)$.

References

dos Santos Sousa, A. R. (2024). A wavelet-based method in aggregated functional data analysis. Monte Carlo Methods and Applications, 30(1), 19-30.

Examples

```
functional_calibration_wavelets(simulated_data$data, simulated_data$weights)
functional_calibration_wavelets(simulated_data$data, simulated_data$weights,
                                tau = 5, p = 0.95, sigma = 0.1, x = simulated_data$x)
functional_calibration_wavelets(simulated_data$data, simulated_data$weights,
                                method = "universal")
```

plot_aggregated_curve *Aggregated Curve Plot*

Description

Generates the plot of the aggregated curve based on the functional coefficients and their corresponding weights. The aggregated curve is computed as:

$$A(x) = \sum_{l=1}^{L} y_l \alpha_l(x)$$

Usage

```
plot_aggregated_curve(alpha, weights, title = NULL, x = NULL)
```

Arguments

`alpha`	A numeric matrix where each column represents the values of a function $\alpha_l(x)$ evaluated at each point in x.
`weights`	A numeric vector with the weight values corresponding to each function $\alpha_l(x)$.
`title`	A string specifying the title of the plot.
`x`	A numeric vector of values at which the function is evaluated. If `NULL`, the default is the sequence `1:nrow(alpha)`.

Value

The function returns the plot of the aggregated function.

Examples

```
plot_aggregated_curve(simulated_data$alphas, c(0.7, 0.3))
plot_aggregated_curve(simulated_data$alphas, c(0.7, 0.3),
                      "Aggregated Curve Example", simulated_data$x)
```

`simulated_data` *Simulated Data*

Description

This is a simulated dataset designed to illustrate the functionalities of the package. It contains 100 samples of aggregated data generated from two functions, $\alpha_1(x)$ and $\alpha_2(x)$, with added Gaussian noise $N(0, 0.1)$.

The functions used in the simulation are:

$$\alpha_1(x) = \sin(5x)e^{-x^2} \quad \alpha_2(x) = \begin{cases} -2, & x < 0 \\ 0, & 0 \leq x < 1.5 \\ 3, & x \geq 1.5 \end{cases}$$

The simulations were performed over an equally spaced grid of 1024 points in the interval [-1, 2]. These functions were linearly combined using random concentrations to generate the samples, with the addition of Gaussian noise.

Usage

```
simulated_data
```

Format

An object of class `list` of length 4.

Value

`data` A data frame with 1024 rows and 100 columns.
Each column represents one sample of the aggregated functions with Gaussian noise $N(0, 0.1)$.

`weigths` A data frame with 2 rows and 100 columns.
Each column contains the random concentrations used to aggregate the two functions in each sample.

`x` A numeric vector of length 1024.
The grid of x-values used in the simulation, equally spaced from -1 to 2.

`alphas` A data frame with 1024 rows and 2 columns.
The true values of the functions $\alpha_1(x)$ and $\alpha_2(x)$ evaluated over the x grid.

`weight_estimation` *Weight Estimation*

Description

Estimates the weights associated with the functional coefficients $\alpha_l(x)$ using the using Ordinary Least Squares.

The problem can be formulated as:

$$A(x) = \sum_{l=1}^{L} y_l \alpha_l(x)$$

where $A(x)$ is the aggregated function evaluated at each point x, $\alpha_l(x)$ are the functional coefficients, and y_l are the weights to be estimated.

Usage

```
weight_estimation(data, alpha)
```

Arguments

`data`	A numeric vector representing one sample of the aggregated function $A(x)$, evaluated at a grid of points x.
`alpha`	A numeric matrix where each column represents the values of a function $\alpha_l(x)$ evaluated at the same grid of points as `data`.

Value

The function returns a vector with the estimated weights obtained using Ordinary Least Squares.

Examples

```
weight_estimation(simulated_data$data[,1], simulated_data$alphas)
```

Index

∗ **datasets**
    simulated_data, 5

functional_calibration_splines, 2
functional_calibration_wavelets, 3

plot_aggregated_curve, 4

simulated_data, 5

weight_estimation, 6