Package: mrct (via r-universe)

August 27, 2024

Type Package

Title Outlier Detection of Functional Data Based on the Minimum Regularized Covariance Trace Estimator

Version 0.0.1.0

Maintainer Jeremy Oguamalam <jeremy.oguamalam@tuwien.ac.at>

Description Detect outlying observations in functional data sets based on the minimum regularized covariance trace (MRCT) estimator. Includes implementation of Oguamalam et al. (2023) <arXiv:2307.13509>.

License GPL (>= 2)

Depends R (>= 4.2.0)

Encoding UTF-8

Imports fda, fdapace, ggplot2, Rdpack, reshape2, robustbase, stats

RdMacros Rdpack

RoxygenNote 7.2.3

NeedsCompilation no

Author Jeremy Oguamalam [aut, cre], Una Radojičić [aut], Peter Filzmoser [aut]

Repository CRAN

Date/Publication 2023-08-17 15:22:43 UTC

Contents

`innerProduct` *Pairwise inner product for L^2 functions*

Description

Calculate all pairwise inner products between elements from L^2 supplied to this function. The integral is approximated by the Trapezoidal rule for uniform grids:

$$\int_a^b f(x)dx \approx \Delta x \left(\sum_{i=1}^{N-1} f(x_i) + \frac{f(x_N) - f(x_0)}{2} \right)$$

whereas $\{x_i\}$ is an uniform grid on $[a, b]$ such that $a = x_0 < x_1 < \ldots < x_N = b$ and Δx the step size, i.e. $\Delta x := x_2 - x_1$. Therefore, it is assumed that the functions are evaluated at the same, equidistant grid.

Usage

```
innerProduct(grid, data)
```

Arguments

`grid`	A numeric vector of the uniform grid on which the functions are evaluated.
`data`	A numeric matrix. Each function has to be a vector stored in a column of `data` and evaluated at the points of `grid`. Thus, the number of rows and columns of `data` correspond to `length(grid)` and the number of functions, respectively.

Value

Numeric symmetric matrix containing the approximated pairwise inner products between the functions supplied by `data`. The entry (i, j) of the result is the inner product between the i-th and j-th column of `data`.

Examples

```
# Create orthogonal fourier basis via `fdapace` package
library(fdapace)
basis <- fdapace::CreateBasis(K = 10,
                              type = "fourier")
iP <- innerProduct(grid = seq(0, 1, length.out = 50), # default grid in CreateBasis()
                   data = basis)
round(iP,3)
# Since the basis is orthogonal, the resulting matrix will be the identity matrix.
```

mrct *Minimum regularized covariance trace estimator*

Description

Functional outlier detection based on the `minimum regularized covariance trace` estimator (Oguamalam et al. 2023) as a robust covariance estimator. This estimator uses a generalization of the Mahalanobis distance for the functional setting (Berrendero et al. 2020) and a corresponding theoretical cutoff value.

Usage

```
mrct(
  data,
  h = 0.75,
  alpha = 0.01,
  initializations = 5,
  subset.iteration = 10,
  seed = 123,
  scaling.iterations = 10,
  scaling.tolerance = 10^(-4),
  criterion = "sum",
  sum.percentage = 0.75
)
```

Arguments

`data` Numeric matrix of a functional data set for which the esimator has to be calculated. Each row contains an observation. They are assumed to be observed on the same regular grid.

`h` Numeric value between 0.5 and 1. Ratio of the data which the estimator is based on. Default is set to 0.75, i.e. 75% of the data will be used for the estimator.

`alpha` Numeric (default is 0.01). Tikhonov regularization parameter α.

`initializations`
Integer (default is 5). Number of random initial subsets.

`subset.iteration`
Integer (default is 10). Maximum number of how often each subset is re-estimated and adjusted.

`seed` Integer (default is 123). Random seed for reproducibility.

`scaling.iterations`
Integer (default is 5). The maximum number of times k_1 is re-scaled if the error between subsequent scalingparameters does not fall below `scaling.tolerance`.

`scaling.tolerance`
Numeric (default is 10^{-4}). The error tolerance for re-scaling. If the error falls below this value, the re-scaling procedure stops.

`criterion` Character. Criterion based on which the optimal subset is chosen among the final subsets. Possible options are: "`cluster`" and the default "`sum`".

`sum.percentage` Numeric value between 0.5 and 1. Corresponding to the "`sum`" criterion. Determines the fraction of observations up to which the sum over the sorted functional Mahalanobis distances is calculated (in ascending order). Default is set to 0.75, i.e. the sum of the smallest 75% of Mahalanobis distances is calculated. If outliers are present, this value should not be to high, in order not to include any outlying curves.

Value

A list:

`theoretical` Integer vector of the indices corresponding to the outliers based on the MRCT estimator.

`theoretical.w` Same as `theoretical` with an additional re-weighting step.

`aMHD` Numeric vector containing the functional Mahalanobis distances of all observations based on the MRCT estimator.

`aMHD.w` Same as `aMHD` with an additional re-weighting step.

`quant` Numeric. Theoretical cutoff value for outlier detection.

`quant.w` Same as `quant` with an additional re-weighting step.

`k` Numeric. Scalingparameter k_1 of Algorithm 1 described in (Oguamalam et al. 2023).

`k.w` Same as `k` with an additional re-weighting step.

`optimal.subset` Integer vector of the optimal h-subset.

`subsets` Numeric matrix containing all final subsets. Each row of `subsets` is one final subset.

`objval` Numeric vector with the objective values of the final subsets based on `criterion`.

References

Berrendero JR, Bueno-Larraz B, Cuevas A (2020). “On Mahalanobis Distance in Functional Settings.” *J. Mach. Learn. Res.*, **21**(9), 1–33..

Oguamalam J, Radojičić U, Filzmoser P (2023). “Minimum regularized covariance trace estimator and outlier detection for functional data.” https://doi.org/10.48550/arXiv.2307.13509..

Examples

```
# Fix seed for reproducibility
set.seed(123)

# Sample outlying indices
cont.ind <- sample(1:50, size=10)

# Generate 50 curves on the interval [0,1] at 50 timepoints with 20% outliers
y <- mrct.rgauss(x.grid=seq(0,1,length.out=50), N=50, model=1,
```

```
                  outliers=cont.ind, method="linear")

# Visualize curves (regular curves grey, outliers black)
colormap <- rep("grey",50); colormap[cont.ind] <- "black"
matplot(x=seq(0,1,length.out=50), y=t(y), type="l", lty="solid",
        col=colormap, xlab="t",ylab="")

# Run MRCT
mrct.y <- mrct(data=y, h=0.75, alpha=0.1,
               initializations=10, criterion="sum")

# Visualize alpha-Mahalanobis distance with cutoff (horizontal black line)
# Colors correspond to simulated outliers, shapes to estimated (MRCT) ones
# (circle regular and triangle irregular curves)
shapemap <- rep(1,50); shapemap[mrct.y$theoretical.w] <- 2
plot(x=1:50, y=mrct.y$aMHD.w, col=colormap, pch=shapemap,
     xlab="Index", ylab=expression(alpha*"-MHD"))
abline(h = mrct.y$quant.w)

# If you dont have any information on possible outliers,
# alternatively you could use the S3 method plot.mrct()
mrct.plot(mrct.y)
```

`mrct.ise` *Integrated square error*

Description

Calculates the approximation of the integrated square error between the estimated covariance based on non-outlying curves of a data set determined by the MRCT estimator and the true kernel for one of the three outlier settings in the simulation study of Oguamalam et al. 2023.

Usage

```
mrct.ise(data, outliers.est, model)
```

Arguments

`data`	Numeric matrix of a functional data set for which the esimator has to be calculated. Each row contains an observation. They are assumed to be observed on the same regular grid.
`outliers.est`	Integer vector containing the indices of outliers.
`model`	Integer. 1 correspond to the first outlier setting, whereas 2 and 3 are related to the remaining two, which both have the same kernel.

Value

Numeric value containing the approximated integrated square error between estimated and theoretical covariance.

References

Oguamalam J, Radojičić U, Filzmoser P (2023). "Minimum regularized covariance trace estimator and outlier detection for functional data." https://doi.org/10.48550/arXiv.2307.13509..

Examples

```
# Fix seed for reproducibility
set.seed(124)

# Sample outlying indices
cont.ind <- sample(1:100,size=10)

# Generate 100 curves on the interval [0,1] at 150 timepoints with 20% outliers.
y <- mrct.rgauss(x.grid=seq(0,1,length.out=150), N=100, model=1,
                 outliers=cont.ind, method="linear")
# Run MRCT
mrct.y <- mrct(data=y, h=0.75, alpha=0.1,
               initializations=10, criterion="sum")
# Two additional curves are regarded as outlying according to the algorithm
mrct.y$theoretical.w %in% cont.ind
# Compare the ISE between true kernel and 1) true non-outliers,
# 2) estimated non-outliers and 3) the complete data
ise1 <- mrct.ise(data=y, outliers.est=cont.ind, model=1)
ise2 <- mrct.ise(data=y, outliers.est=mrct.y$theoretical.w, model=1)
ise3 <- mrct.ise(data=y, outliers.est=c(), model=1)
ise1; ise2; ise3
```

---

mrct.plot — *Plot function for result from* `mrct()`

Description

A function for descriptive plots for an object resulting from a call to `mrct()`.

Usage

```
mrct.plot(mrct.object)
```

Arguments

`mrct.object`	A result from a call to `mrct()`.

Value

Descriptive plots

`aMHD.plot`	Alpha-Mahalanobis distances, corresponding cutoff values and coloring according to estimated outliers (grey regular, black irregular).
`aMHD.plot.w`	Same as `aMHD.plot`, with additional re-weighting step.

Examples

```
# Similar to example in mrct() helpfile
# Fix seed for reproducibility
set.seed(123)

# Sample outlying indices
cont.ind <- sample(1:50, size=10)

# Generate 50 curves on the interval [0,1] at 50 timepoints with 20% outliers
y <- mrct.rgauss(x.grid=seq(0,1,length.out=50), N=50, model=1,
                 outliers=cont.ind, method="linear")

# Visualize curves (regular curves grey, outliers black)
colormap <- rep("grey",50); colormap[cont.ind] <- "black"
matplot(x=seq(0,1,length.out=50), y=t(y), type="l", lty="solid",
        col=colormap, xlab="t",ylab="")

# Run MRCT
mrct.y <- mrct(data=y, h=0.75, alpha=0.1,
               initializations=10, criterion="sum")

# Visualize alpha-Mahalanobis distance
# Colorinfromation according to estimated outliers (grey regular, black irregular)
mrct.plot(mrct.y)
```

`mrct.rgauss` *Random sample from Gaussian process*

Description

Generate random samples of Gaussian process on a uniform grid for different settings of the simulation study in Oguamalam et al. 2023.

Usage

```
mrct.rgauss(
  x.grid,
  N,
  seed = 123,
  model,
  outliers,
  sigma = 1,
  l = 1,
  method = "linear"
)
```

Arguments

x.grid	Numeric vector containing a uniform grid on which the process is defined.
N	Integer number of observations to generate.
seed	Integer (default is 123).. Random seed for reprocudibility.
model	Integer. Either $1, 2$ or 3. Corresponds to one of the three simulation settings.
outliers	Integer vector containing the indices of outliers. If empty, then only regular curves will be generated.
sigma, l	Numeric values with default equal to 1. Parameters for the covariance kernel.
method	Different types of covariance kernels. Possible options are "quadratic" $$\gamma(s,t) = \sigma \text{exp}\left(\frac{-(s-t)^2}{l}\right),$$ "linear" $$\gamma(s,t) = \sigma \text{exp}\left(\frac{-\lvert s-t \rvert}{l}\right)$$ or "gaussian" (default) $$\gamma(s,t) = \sigma^2 \text{exp}\left(\frac{-(s-t)^2}{2l^2}\right)$$.

Value

Numeric matrix with N rows and `length(x.grid)` columns containing the randomly generated curves following a Gaussian process. Each observations is a row of the result.

References

Oguamalam J, Radojičić U, Filzmoser P (2023). "Minimum regularized covariance trace estimator and outlier detection for functional data." https://doi.org/10.48550/arXiv.2307.13509..

Examples

```
# Fix seed for reproducibility
set.seed(123)

# Sample outlying indices
cont.ind <- sample(1:50,size=10)

# Generate 50 curves on the interval [0,1] at 50 timepoints with 20% outliers
y <- mrct.rgauss(x.grid=seq(0,1,length.out=50), N=50 ,model=1,
                 outliers=cont.ind)

# Visualize curves (regular curves grey, outliers black)
colormap <- rep("grey",50); colormap[cont.ind] <- "black"
matplot(x=seq(0,1,length.out=50), y=t(y), type="l", lty="solid",
        col=colormap, xlab="t",ylab="")
```

mrct.sparse — *Sparse minimum regularized covariance trace estimator*

Description

Robust outlier detection for sparse functional data as a generalization of the `minimum regularized covariance trace` (MRCT) estimator (Oguamalam et al. 2023). At first the observations are smoothed by a B-spline basis and afterwards the MRCT algorithm is performed with the matrix of basis coefficients.

Usage

```
mrct.sparse(
  data,
  nbasis = dim(data)[2],
  new.p = dim(data)[2],
  h = 0.75,
  alpha = 0.01,
  initializations = 5,
  seed = 123,
  scaling.iterations = 10,
  scaling.tolerance = 10^(-4),
  criterion = "sum",
  sum.percentage = 0.75
)
```

Arguments

`data`	Numeric matrix of a functional data set for which the esimator has to be calculated. Each row contains an observation. They are assumed to be observed on the same (probably sparse) regular grid. The number of grid points must be at least `nbasis`.
`nbasis`	Integer. Number of B-spline basis functions for smoothing. The basis will be of order 4 and therefore, cannot contain less than 4 functions. The default value will be set to `dim(data)[2]`. i.e. the number of time points with a maximum of 15.
`new.p`	Integer. Length of the grid of the smoothed curves. The resulting grid will be an equidistant partition of `[rangeval[1],rangeval[length(rangeval)]]`. Default value is `dim(data)[2]`
`h`	Numeric value between 0.5 and 1. Ratio of the data which the estimator is based on. Default is set to 0.75, i.e. 75% of the data will be used for the estimator.
`alpha`	Numeric (default is 0.01). Tikhonov regularization parameter α.
`initializations`	Integer (default is 5). Number of random initial subsets.
`seed`	Integer (default is 123). Random seed for reproducibility.

scaling.iterations
: Integer (default is 5). The maximum number of times k_1 is re-scaled if the error between subsequent scalingparameters does not fall below `scaling.tolerance`.

scaling.tolerance
: Numeric (default is 10^{-4}). The error tolerance for re-scaling. If the error falls below this value, the re-scaling procedure stops.

criterion
: Character. Criterion based on which the optimal subset is chosen among the final subsets. Possible options are: "cluster" and the default "sum".

sum.percentage
: Numeric value between 0.5 and 1. Corresponding to the "sum" criterion. Determines the fraction of observations up to which the sum over the sorted functional Mahalanobis distances is calculated (in ascending order). Default is set to 0.75, i.e. the sum of the smallest 75% of Mahalanobis distances is calculated. If outliers are present, this value should not be to high, in order not to include any outlying curves.

Value

A list with two entries

mrct.output
: List. The same output as the function `mrct()`. For more details, see there.

data.smooth
: Numeric matrix. Collection of the smoothed curves of `data` with `dim(data)[1]` rows and `new.p` columns. Each row corresponds to one observation.

References

Oguamalam J, Radojičić U, Filzmoser P (2023). "Minimum regularized covariance trace estimator and outlier detection for functional data." https://doi.org/10.48550/arXiv.2307.13509..

Examples

```
# Fix seed for reproducibility
set.seed(123)

# Sample outlying indices
cont.ind <- sample(1:50,size=10)

# Generate 50 sparse curves on the interval [0,1] at 10 timepoints with 20% outliers
y <- mrct.rgauss(x.grid=seq(0,1,length.out=10), N=50, model=1,
                 outliers=cont.ind, method="linear")

# Visualize curves (regular curves grey, outliers black)
colormap <- rep("grey",50); colormap[cont.ind] <- "black"
matplot(x = seq(0,1,length.out=10), y = t(y), type="l", lty="solid",
        col=colormap, xlab="t",ylab="")

# Run sparse MRCT
sparse.mrct.y <- mrct.sparse(data = y, nbasis = 10, h = 0.75, new.p = 50,
                             alpha = 0.1, initializations = 10, criterion = "sum" )

# Visualize smoothed functions
```

```
matplot(x=seq(0,1,length.out=50), y=t(sparse.mrct.y$data.smooth),
        type="l", lty="solid", col=colormap, xlab="t", ylab="")

# Visualize alpha-Mahalanobis distance with cutoff (horizontal black line)
# Colors correspond to simulated outliers, shapes to estimated (sparse MRCT) ones
# (circle regular and triangle irregular curves)
shapemap <- rep(1,50); shapemap[sparse.mrct.y$mrct.output$theoretical.w] <- 2
plot(x = 1:50, y = sparse.mrct.y$mrct.output$aMHD.w, col=colormap, pch = shapemap,
     xlab = "Index", ylab = expression(alpha*"-MHD"))
abline(h = sparse.mrct.y$mrct.output$quant.w)

# If you dont have any information on possible outliers,
# alternatively you could use the S3 method plot.mrctsparse()
mrct.sparse.plot(mrct.sparse.object = sparse.mrct.y)
```

`mrct.sparse.plot` *Plot function for result from* `mrct.sparse()`

Description

A function for descriptive plots for an object resulting from a call to `mrct.sparse()`.

Usage

```
mrct.sparse.plot(
  x = seq(0, 1, length.out = dim(mrct.sparse.object[[2]])[2]),
  mrct.sparse.object
)
```

Arguments

`x` Gridpoints on which the smoothed data is to be plotted on. Default is `seq(0,1,length.out=new.p)` whereas `new.p` is a parameter set in the call to `mrct.sparse()`.

`mrct.sparse.object` A result from a call to `mrct.sparse()`.

Value

Descriptive plots.

`aMHD.plot` Alpha-Mahalanobis distances, corresponding cutoff values and coloring according to estimated outliers (grey regular, black irregular).

`aMHD.plot.w` Same as `aMHD.plot`, with additional re-weighting step.

`data.plot` Plot of the smoothed curves, colors corresponding to estimated outliers (gery regular, black irregular). Per default, the x-axis is plotted over an equidistant grid of the interval $[0, 1]$.

Examples

```
# Similar to example in mrct.sparse() helpfile
# Fix seed for reproducibility
set.seed(123)

# Sample outlying indices
cont.ind <- sample(1:50,size=10)

# Generate 50 sparse curves on the interval [0,1] at 10 timepoints with 20% outliers
y <- mrct.rgauss(x.grid=seq(0,1,length.out=10), N=50, model=1,
                 outliers=cont.ind, method="linear")

# Visualize curves (regular curves grey, outliers black)
colormap <- rep("grey",50); colormap[cont.ind] <- "black"
matplot(x = seq(0,1,length.out=10), y = t(y), type="l", lty="solid",
        col=colormap, xlab="t",ylab="")

# Run sparse MRCT
sparse.mrct.y <- mrct.sparse(data = y, nbasis = 10, h = 0.75, new.p = 50,
                             alpha = 0.1, initializations = 10, criterion = "sum" )

# Visualize alpha-Mahalanobis distances and smoothed curves
# Colorinformation according to estimated outliers (grey regular, black irregular)
mrct.sparse.plot(mrct.sparse.object = sparse.mrct.y)
```

Index