Package ‘LMD’

October 12, 2022

Type Package
Date 2022-09-10
Title A Self-Adaptive Approach for Demodulating Multi-Component Signal
Version 1.0.0
Maintainer Shubhra Prakash <shubhraprakash279@gmail.com>
Description Local Mean Decomposition is an iterative and self-adaptive approach for demodulating, processing, and analyzing multi-component amplitude modulated and frequency modulated signals. This R package is based on the approach suggested by Smith (2005) <doi:10.1098/rsif.2005.0058> and the 'Python' library 'PyLMD'.
License Apache License (>= 2)
Depends R (>= 3.6.0)
BugReports https://github.com/shubhra-opensource/LMD/issues
URL https://github.com/shubhra-opensource/LMD
Encoding UTF-8
RoxygenNote 7.2.1
Suggests knitr, rmarkdown, ggformula, testthat (>= 3.0.0)
Config/testthat/edition 3
VignetteBuilder knitr
Imports EMD, ggplot2, patchwork
NeedsCompilation no
Author Shubhra Prakash [trl, aut, cre]
Repository CRAN
Date/Publication 2022-09-20 09:56:07 UTC

R topics documented:

extract_product_function ............................................ 2
find_extrema .................................................................. 3
is_monotonous ............................................................... 4
**extract_product_function**

**Extract Product Function**

**Description**

Method for extracting product functions

**Usage**

```r
extract_product_function(
  signal,
  max_envelope_iteration = 200,
  envelope_epsilon = 0.01,
  convergence_epsilon = 0.01
)
```

**Arguments**

- `signal`  
  Signal values (Numeric | vector)
- `max_envelope_iteration`  
  Maximum number of iterations when separating local envelope signals (Integer)
- `envelope_epsilon`  
  Terminate processing when obtaining pure FM signal (Double)
- `convergence_epsilon`  
  Terminate processing when modulation signal converges (Double)

**Value**

Product Function

**Author(s)**

Shubhra Prakash, <shubhraprakash279@gmail.com>

**References**

[https://pypi.org/project/PyLMD/](https://pypi.org/project/PyLMD/)
find_extrema

Examples

```r
x=1:100
y = (2 / 3) * sin(x * 30) + (2 / 3) * sin(x * 17.5) + (4 / 5) * cos(x * 2)
plot(y,type="l")
pf=extract_product_function(y)
```

find_extrema  Find Extreme Points

Description

Method for finding Extreme Points

Usage

```r
find_extrema(signal, include_endpoints = TRUE)
```

Arguments

- **signal**: Signal values (Numeric | vector)
- **include_endpoints**: whether to include end points or not (Boolean)

Details

A local extrema is the point at which a maximum or minimum value of the function in some open interval containing the point is obtained.

Value

Indexes of all extrema values (including starting and ending points)

Author(s)

Shubhra Prakash, <shubhraprakash279@gmail.com>

Examples

```r
signal=c( 0.841471 ,0.9092974,0.14112,-0.7568025,-0.9589243)
find_extrema(signal)
```
Description

Method for checking if signal is increasing or decreasing monotonously

Usage

is_monotonous(signal)

Arguments

signal Signal values (Numeric | vector)

Details

A monotonic signal is a function that keeps increasing or decreasing as its domain variable proceeds.

Value

Boolean

Author(s)

Shubhra Prakash, <shubhraprakash279@gmail.com>

References

https://pypi.org/project/PyLMD/

Examples

x=1:100
is_monotonous(x)
**lmd**

**Local Mean Decomposition**

**Description**

Method for finding Product Functions (PFs)

**Usage**

```r
lmd(
    signal,
    include_endpoints = TRUE,
    max_smooth_iteration = 12,
    max_envelope_iteration = 200,
    envelope_epsilon = 0.01,
    convergence_epsilon = 0.01,
    max_num_pf = 8
)
```

**Arguments**

- **signal**: Signal values (Numeric | vector)
- **include_endpoints**: Whether to treat the endpoint of the signal as a pseudo-extreme point (Boolean)
- **max_smooth_iteration**: Maximum number of iterations of moving average algorithm (Integer)
- **max_envelope_iteration**: Maximum number of iterations when separating local envelope signals (Integer)
- **envelope_epsilon**: Terminate processing when obtaining pure FM signal (Double)
- **convergence_epsilon**: Terminate processing when modulation signal converges (Double)
- **max_num_pf**: The maximum number of PFs generated (Integer)

**Details**


**Value**

```
list(pf,residue) | PFs:The decompose functions arranged from high frequency to low frequency | residue:residual component
```
local_mean_and_envelope

Author(s)
Shubhra Prakash, <shubhraprakash279@gmail.com>

References
https://pypi.org/project/PyLMD/

Examples
```
x=1:100
y = (2 / 3) * sin(x * 30) + (2 / 3) * sin(x * 17.5) + (4 / 5) * cos(x * 2)
plot(y,type="l")
lmd(y)
```

local_mean_and_envelope

(Local Mean and Envelope)

Description
Method for finding Local Mean and Envelope

Usage
```
local_mean_and_envelope(signal, extrema)
```

Arguments
- `signal` Signal values (Numeric | vector)
- `extrema` indexes for extreme values

Value
mean, envelope and smoothed mean and envelope values

Author(s)
Shubhra Prakash, <shubhraprakash279@gmail.com>

References
https://pypi.org/project/PyLMD/

Examples
```
signal = sin(1:10)
extrema = c(1, 2, 5, 8, 10)
local_mean_and_envelope(signal, extrema)
```
Weighted Moving Average

**Description**

Weighted Moving Average Smoothing

**Usage**

```r
moving_average_smooth(signal, window, max_smooth_iteration = 12)
```

**Arguments**

- `signal`: Signal values (Numeric | vector)
- `window`: Filter weights for smoothing (Numeric | vector)
- `max_smooth_iteration`: Maximum number of iterations of moving average algorithm (Integer)

**Details**

Weighted Moving Average Smoothing is used to smooth the mean and envelope signal

**Value**

Smooth signal

**Author(s)**

Shubhra Prakash, <shubhaprakash279@gmail.com>

**References**

https://pypi.org/project/PyLMD/

**Examples**

```r
x=0:100
y = (2 / 3) * sin(x * 30) + (2 / 3) * sin(x * 17.5) + (4 / 5) * cos(x * 2)
plot(y,type="l")
wma=moving_average_smooth(y,5)
plot(wma,type="l")
```
plot_lmd  

*LMD Plot*

**Description**

Method for plotting Product Functions (PFs) and Residue

**Usage**

```r
plot_lmd(
  lmd_obj,
  max_pf = length(lmd_obj[["pf"]]),
  show_residue = TRUE,
  pricolor_plot = "midnightblue",
  line_size_plot = 1
)
```

**Arguments**

- `lmd_obj` LMD object created from LMD function
- `max_pf` Number of PFs to Plot
- `show_residue` Whether to plot residue or not
- `pricolor_plot` color of plots
- `line_size_plot` Size of line in ggplot

**Value**

ggplot plot for Product Functions (PFs) and Residue

**Author(s)**

Shubhra Prakash, <shubhraprakash279@gmail.com>

**Examples**

```r
x=1:100
y = (2 / 3) * sin(x * 30) + (2 / 3) * sin(x * 17.5) + (4 / 5) * cos(x * 2)
plot_lmd(lmd(y))
```
Index

* Average
  moving_average_smooth, 7
* Demodulation
  lmd, 5
* Envelope
  local_mean_and_envelope, 6
* Fault
  lmd, 5
* LMD
  plot_lmd, 8
* Local
  lmd, 5
  local_mean_and_envelope, 6
* Mean
  local_mean_and_envelope, 6
* Moving
  moving_average_smooth, 7
* PF
  plot_lmd, 8
* Product
  lmd, 5
* Residue
  plot_lmd, 8
* Weighted
  moving_average_smooth, 7
* and
  local_mean_and_envelope, 6
* decomposition
  lmd, 5
* diagnosis
  lmd, 5
* extrema
  find_extrema, 3
* functions
  lmd, 5
* lmd
  extract_product_function, 2
  lmd, 5
* mean
  lmd, 5
* mechanism
  lmd, 5
* monotonic
  is_monotonous, 4
* monotonous
  is_monotonous, 4
* self-adaptive
  lmd, 5
* wma
  moving_average_smooth, 7
  extract_product_function, 2
  find_extrema, 3
  is_monotonous, 4
  lmd, 5
  local_mean_and_envelope, 6
  moving_average_smooth, 7
  plot_lmd, 8