

Package: FARM (via r-universe)

October 1, 2026

Title Forward Angular Relevance Measure

Version 0.1.5.15

Description The original implementation of the Forward Angular Relevance Measure (FARM). The algorithm relies on a forward alignment approach for compared time series based on the dynamic time warping (DTW) principle. It considers the differences between data point as source in a combined distance metric that is used for series alignment. The algorithm returns a global and a series of local relevance measures relying on correlation coefficients. A normalization of time series is not part of the algorithm but recommended for best results. The FARM method is introduced in: Christen et al. (2023) [doi:10.48550/arXiv.2304.11028](https://doi.org/10.48550/arXiv.2304.11028).

License GPL-3

Encoding UTF-8

Imports dplyr, stats, utils

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Ramón Christen [aut, cre] (ORCID: <https://orcid.org/0000-0002-3963-0785>)

Maintainer Ramón Christen <hiroshima@bluewin.ch>

Repository <https://amishorn.r-universe.dev>

Date/Publication 2026-09-30 13:54:42 UTC

RemoteUrl <https://github.com/cran/FARM>

RemoteRef HEAD

RemoteSha 52f725f94cc0f33e5ff144113a2566955db14302

Contents

farm	2
farm.dist	3

farm.mean.p	4
farm.slm.f	5
tw.decomp	6

Index	7
--------------	----------

farm	<i>Forward Angular Relevance Metric (FARM)</i>
------	--

Description

farm identifies and compensates time warping between query and reference time series by sample and hold and a deletion for a remove. Subsequent, the algorithm calculates two relevance values: (1) the relevance of the query time series to the reference time series for each point locally in the given time window for local relevance and (2) the mean of a proportion with highest local relevance values and the mean of all local relevance values. The local relevance variation of any local sequence (i.e. local query or reference sequence) equals 0 (var(xts) = 0) if the sequence is a constant series with a std. variation of 0.000001 to normalized (mean = 0) sequences. This avoids NA in constant series.

Usage

```
farm(  
  refTS,  
  qryTS,  
  lcwin = 5,  
  rel.th = 15,  
  ff.align = TRUE,  
  reshape.fnc = function(x) x,  
  fuzzyc = c(2, 6, 10, 6, 2)/10,  
  metric.space = TRUE,  
  reshape.mode = "uni"  
)
```

Arguments

refTS	double vector: reference time series
qryTS	double vector: query time series
lcwin	int: length of local correlation window - default: 5
rel.th	int: proportion of local relevance measure with highest values included in global measure - default: 15
ff.align	boolean: forward feature alignment. If set to FALSE, the algorithm calculates relevance on originally passed time series without prior feature alignment. - default: TRUE
reshape.fnc	function(x): applied function on calculated local relevance values that are multiplied with qryTS for reshaping (suppressing irrelevant features). For threshold reshaping (i.e. shap.series = orig IF rel >= th ELSE 0) set the function to: function(x) (x >= th)*1 - default: function(x) x

fuzzyc	double vector: fuzzy coefficients. Ideally, $\text{length}(\text{fuzzyc}) = \text{lcwin}$. To disable fuzzification, set $\text{fuzzyc} = 1$ - default: $c(2,6,10,6,2)/10$
metric.space	boolean: triangle inequality compliance flag. If TRUE, the alignment algorithm only applies the sine function for distance instead of the enhanced sine-exp function composition. - default: TRUE
reshape.mode	keyword {'uni', 'dual'}: mode for decomposing. uni: decomposing only query time series. dual: decomposing query and reference time series. In the dual mode, decomposing only inserts data by interpolation but does not delete any information (no information loss). default: uni

Value

```
list(path, rts.decomp, qts.decomp, qts.shaped, rel.local, rel.local.fuzz, rel.global, qmeas.mean.p,
      path, warping path,
      rts.decomp, time warping decomposed reference time series,
      qts.decomp, time warping decomposed query time series,
      qts.shaped, shaped dewarped (if ff.align = T) query time series according to the reshape factor,
      rel.local, local relevance values (length(rel.local) = length(qryTS)) compliant with metric.space,
      rel.local.end, local relevance values (not fuzzyfied) with correlation coefficient spot to spot,
      rel.local.fuzz, local relevance values fuzzyfied according to fuzzyc factors applied on rel.local,
      rel.global, global relevance value (scalar) based on local relevance values rel.local (best fit),
      qmeas.mean.p, mean absolute product quality measure for alignment of query with respect to reference,
      qmeas.slm.f, slope-mean-factorial slm_f quality measure (mean product of slopes) for alignment
```

Note

date: 11-17-2022 farm

Author(s)

rch

Examples

```
farm(c(1,2,2,1,3), c(1,1,2,1,4), lcwin = 3, rel.th = 10)
```

farm.dist

FARM distance

Description

The returned distance is the sine of the absolute angle between vectors if both have a positive or both have a negative argument. For contrasting vectors (v1: +-x,+y and v2: +-x,-y) the resulting distance is: $1 - \exp(-\phi * 5)$, for non-metric space and: . The vector argument 0 is treated as positive argument.

Usage

```
farm.dist(dyr, dyq = NA, metric.space = FALSE)
```

Arguments

dyr	double: dy reference normalized to dx = 1
dyq	double: dy query normalized to dx = 1. Not required if dyr also comprises query delta values in a data frame.
metric.space	boolean: triangle inequality compliance flag. If TRUE, the distance relies on the sine function only instead of the enhanced sine-exp function composition. - default: FALSE

Value

double: distance value

Note

date: 11-17-2022 farm.dist

Author(s)

rch

Examples

```
farm.dist(0.23, -2.85)
```

farm.mean.p

farm.mean.p

Description

An absolute values based quality measure for feature alignment of two independent time series with the **same** length. The measure returns the mean of the absolute product of two time series according the formula:

$$\left| \frac{1}{n} \sum_{i=0}^n ref_i \cdot qry_i \right|$$

Resulting from the product, the **higher** the values, the **better** the result.

Usage

```
farm.mean.p(rts, qts)
```

Arguments

rts double vector: reference time series of length n
qts double vector: query time series of length n

Value

double: quality measure *larger means better*

Examples

```
farm.mean.p(c(1,2,1.5,2.1,5,2.32,1,0.2), c(0.2,0.4,0.4,0.5,1,0.92,0.5,0.2))
```

farm.slm.f	<i>farm.slm.f</i>
------------	-------------------

Description

The slope-mean-factorial `slm_f` returns a quality measure for feature alignment of two independent time series with the **same** length. The measure relies on the mean product of the slopes (i.e. differences) of subsequent data values in two time series, according the formula:

$$\frac{\frac{1}{n} \sum_{i=2}^n \Delta ref_{[i-1,i]} \cdot \Delta qry_{[i-1,i]}}{1 + \left| \frac{\#smp_{orig} - \#smp_{method}}{\#smp_{orig}} \right|}$$

Resulting from the inner product of the slopes, the **higher** the values, the **better** the result. For penalizing the number of changes in time series for feature matching, the original and modified series length can be set by the parameters `l.orig` and `l.adj` respectively.

Usage

```
farm.slm.f(rts, qts, l.orig = 1, l.adj = 1)
```

Arguments

rts double vector: reference time series of length n
qts double vector: query time series of length n
l.orig int: length of original time series - default: 1
l.adj int: length of adjusted time series - default: 1

Value

double: quality measure *larger means better*

Examples

```
farm.slm.f(c(1,2,1.5,2.1,5,2.32,1,0.2), c(0.2,0.4,0.4,0.5,1,0.92,0.5,0.2))
```

tw.decomp

*FARM time warping decomposition***Description**

Two time series present a warping scheme to each other. For comparison and advanced analysis of warped time series, it requires de-warping time series according a given scheme. This function expands the reference and query time series for resulting in a 1 to 1 data point alignment. The series are expanded according to the farm warping solution by interpolation. There are no data points removed.

Usage

```
tw.decomp(rts, qts, twp, mode = "uni", abs = FALSE)
```

Arguments

rts	vector: original reference time series
qts	vector: original query time series
twp	data.frame: warped data point assignment. df requires an x (qry) and y (ref) variable providing assignment information. e.g. a data point of query ts is assigned to two subsequent reference data points results to: x=[.., x_i, x_i+1, x_i+1, ..] y=[.., y_i, y_i+1, y_i+2, ..]
mode	keyword {'uni', 'dual'}: mode for decomposing. uni: decomposing only query time series. dual: decomposing query and reference time series. In the dual mode, decomposing only inserts data by interpolation but does not delete any information (no information loss). default: uni
abs	bool: true if series are amplitude values (absolute); false if series are delta values of subsequent samples (relative)

Value

```
list(rts=c(rts1, rts2, ..), qts=c(qts1, qts2, ..)): de-warped reference (rts) and query (qts) time series
```

Note

```
date: 11-17-2022 tw.decomp
```

Author(s)

```
rch
```

Examples

```
tw.decomp(c(0,1,0,0,0), c(0,0.5,0.5,0,0), data.frame(x=c(1,2,2,3,4), y=c(1,2,3,4,4)))
```

Index

farm, [2](#)
farm.dist, [3](#)
farm.mean.p, [4](#)
farm.slm.f, [5](#)
tw.decomp, [6](#)