Package ‘FSMUMI’

November 26, 2018

Type Package

Title Imputation of Time Series Based on Fuzzy Logic

Version 1.0

Date 2018-11-16

Author Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

Maintainer Thi Thu Hong Phan <ptthong@vnua.edu.vn>

Description Filling large gaps in low or uncorrelated multivariate time series uses a new fuzzy weighted similarity measure. It contains all required functions to create large missing consecutive values within time series and then fill these gaps, according to the paper Phan et al. (2018), <DOI:10.1155/2018/9095683>. Performance indicators are also provided to compare similarity between two univariate signals (incomplete signal and imputed signal).

Depends R (>= 3.0.0)

Imports FuzzyR, stats, lsa

License GPL (>= 2)

RoxygenNote 6.0.1

URL http://mawenzi.univ-littoral.fr/FSMUMI/

NeedsCompilation no

Repository CRAN

Date/Publication 2018-11-26 20:10:06 UTC

R topics documented:

FSMUMI-package *Imputation of Time Series Based on Fuzzy Logic*

Description

Filling large gaps in low or uncorrelated multivariate time series uses a new fuzzy weighted similarity measure. It contains all required functions to create large missing consecutive values within time series and then fill these gaps, according to the paper Phan et al. (2018), <DOI:10.1155/2018/9095683>. Performance indicators are also provided to compare similarity between two univariate signals (incomplete signal and imputed signal).

Details

Index of help topics:

```
Creating_gap_univariate
                        Creating a gap in a univariate series
Creating_gaps           Creating gaps in multivariate time series
FSMUMI-package          Imputation of Time Series Based on Fuzzy Logic
FSMUMImputation         Imputing large gaps based on the new
                        fuzzy-weighted similarity measure
Indexes_size_missing    Indexes and sizes of gaps
compute.ed              Euclidean distance (ED)
compute.fa2             FA2
compute.fb              Fractional Bias (FB)
compute.fsd             Fraction of Standard Deviation (FSD)
compute.rmse            Root Mean Square Error (RMSE)
compute.sim             Similarity
create.Myfis            Building a fuzzy inference system
dataFSMUMI              An example of a multivariate times series
                        having three signals
evalfis                 FIS evaluation
imp_1NA                 Completing isolated missing points (1NA)
```

Author(s)

Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

Maintainer: Thi Thu Hong Phan <ptthong@vnua.edu.vn>

References

Thi-Thu-Hong Phan, Andre Bigand, and Emilie Poisson Caillault, A New Fuzzy Logic-Based Similarity Measure Applied to Large Gap Imputation for Uncorrelated Multivariate Time Series, Applied Computational Intelligence and Soft Computing, vol. 2018, Article ID 9095683, 15 pages, 2018, <DOI:10.1155/2018/9095683>.

Examples

```
# Load package dataset
data(dataFSMUMI)
X <- dataFSMUMI[1:3000,]


# Create gaps in multivariate time series
rate=0.01 # Percentage of missing values on a signal
ngaps=1 # Number of gaps on each signal
data <- Creating_gaps(X, rate,ngaps)

#Indexes of gaps
ind=Indexes_size_missing(data);

#Imputation parameters tuning
large_gap_threshold= 30
step_threshold=30
step_finding=10

# Fill gaps using FSMUMI algorithm
results_FSMUMI <- FSMUMImputation(data, large_gap_threshold, step_threshold, step_finding)

# Plot true values (black) and imputation values on the first signal
Position_of_gap=max(1,ind[[1]][1,1]):min(ind[[1]][1,1]+ind[[1]][1,2])
plot(Position_of_gap,X[Position_of_gap,1], type = "l",ylab="Value")
lines(Position_of_gap,results_FSMUMI[Position_of_gap,1], col = "red", lty = "dashed")

# Compute the similarity between the first imputed signal and the first real signal
compute.sim(X[,1], results_FSMUMI[,1])

# Compute the RMSE  between the first imputed signal and the first real signal
compute.rmse(X[,1], results_FSMUMI[,1])

# Compute the FA2  between the first imputed signal and the first real signal
compute.fa2(X[,1], results_FSMUMI[,1])

# Compute the FSD  between the first imputed signal and the first real signal
compute.fsd(X[,1], results_FSMUMI[,1])

# Compute the FB  between the first imputed signal and the first real signal
compute.fb(X[,1], results_FSMUMI[,1])
```

compute.ed *Euclidean distance (ED)*

Description

Compute the Euclidean distance between two vectors having the same length Y and X.

Usage

```
compute.ed(Y, X)
```

Arguments

Y	vector of imputed values
X	vector of true values

Details

This function returns the Euclidean distance of two vectors corresponding to univariate signals. A lower ED ($ED \in [0, \inf]$) value indicates that the two vectors are more similar. The both vectors Y and X must be of equal length, on the contrary an error will be displayed. In two input vectors, eventual NA will be exluded with a warning diplayed.

Author(s)

Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

Examples

```
data(dataFSMUMI)
X <- dataFSMUMI[, 1] ; Y <- dataFSMUMI[, 2]
compute.ed(Y,X)
```

compute.fa2 *FA2*

Description

Calculate the FA2 between two univariate signals Y (imputed values) and X (true values).

Usage

```
compute.fa2(Y, X, verbose = F)
```

Arguments

Y	vector of imputed values
X	vector of true values
verbose	if TRUE, print advice about the quality of the model

Details

This function returns the FA2 value which corresponds to the percentage of pairs of values (x_i, y_i) satisfying the condition $0,5 <= (y_i/x_i) <= 2$. FA2 is close to 1, the imputation model is more accurate. Both vectors Y and X must be of equal length, on the contrary an error will be displayed. In both input vectors, eventual NA will be exluded with a warning diplayed.

Author(s)

Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

Examples

```
data(dataFSMUMI)
X <- dataFSMUMI[, 1] ; Y <- dataFSMUMI[, 2]
compute.fa2(Y,X)
compute.fa2(Y,X, verbose = TRUE)

# By definition, if pairs of true and imputed values are zero,
# FA2 corresponding to this pair of values equals 1.
X[1] <- 0
Y[1] <- 0
compute.fa2(Y,X)
```

---

compute.fb *Fractional Bias (FB)*

Description

Calculate the FB between two univariate signals Y (imputed values) and X (true values).

Usage

```
compute.fb(Y, X, verbose = F)
```

Arguments

Y	vector of imputed values
X	vector of true values
verbose	if TRUE, print advice about the quality of the model

Details

This function returns the FB value of two vectors univariate signals. This indicator indicates whether predicted values are underestimated or overestimated compared to true values. A perfect imputation model has $FB = 0$. An imputation model is acceptable when its FB value is less than or equal to 0.3 ($FB <= 0.3$). The two vectors Y and X are the same length, on the contrary an error will be displayed. In both input vectors, eventual NA will be exluded with a warning diplayed.

Author(s)

Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

Examples

```
data(dataFSMUMI)
X <- dataFSMUMI[, 1] ; Y <- dataFSMUMI[, 2]
compute.fb(Y,X)
compute.fb(Y,X, verbose = TRUE)

# If mean(X)=mean(Y)=0, it is impossible to estimate FB,
# unless both true and imputed values vectors are constant.
# By definition, in this case, FB = 0.
X <- rep(0, 10) ; Y <- rep(0, 10)
compute.fb(Y,X)

# If true and imputed values are not zero and are opposed, FB = Inf.
X <- rep(runif(1), 10)
Y <- -X
compute.fb(Y,X)
```

`compute.fsd` *Fraction of Standard Deviation (FSD)*

Description

Compute the Fraction of Standard Deviation (FSD) of two univariate signals Y (imputed values) and X (true values).

Usage

```
compute.fsd(Y, X, verbose = F)
```

Arguments

`Y`	vector of imputed values
`X`	vector of true values
`verbose`	if TRUE, print advice about the quality of the model

Details

This function returns the FSD value between two univariate signals. FSD value approaches zero means that a better performance method for the imputation task. Y and X must have the same length, conversely an error will be appeared. In both input vectors, NA will be exluded with a warning diplayed.

Author(s)

Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

Examples

```
data(dataFSMUMI)
X <- dataFSMUMI[, 1] ; Y <- dataFSMUMI[, 2]
compute.fsd(Y,X)
compute.fsd(Y,X, verbose = TRUE)

# By definition, if true and imputed values are equal and constant,
# FSD = 0.
X <- rep(runif(1), 10)
Y <- X
compute.fsd(Y,X)

# However, if true and imputed values are constant but different,
# FSD is not calculable. An error is displayed.
## Not run:
X <- rep(runif(1), 10);Y <- rep(runif(1), 10)
compute.fsd(Y,X)
## End(Not run)
```

`compute.rmse` *Root Mean Square Error (RMSE)*

Description

Calculate the FA2 between two univariate signals Y (imputed values) and X (true values).

Usage

```
compute.rmse(Y, X)
```

Arguments

Y	vector of imputed values
X	vector of true values

Details

This function computes the value of RMSE of two univariate signals. A lower RMSE ($RMSE \in [0, \inf]$) value demonstrates a better performance method for the imputation task. The length of the two vectors Y and X must be equal, on the contrary an error will be displayed. In both input vectors, eventual NA will be exluded with a warning diplayed.

Author(s)

Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

Examples

```
data(dataFSMUMI)
X <- dataFSMUMI[, 1] ; Y <- dataFSMUMI[, 2]
compute.rmse(Y,X)
```

`compute.sim` *Similarity*

Description

Compute the percentage of similarity of two univariate signals Y (imputed values) and X (true values).

Usage

```
compute.sim(Y, X)
```

Arguments

`Y`	vector of imputed values
`X`	vector of true values

Details

This function returns the value of similarity of two v univariate signals. A higher similarity ($Similarity \in [0, 1]$) highlights a more accurate method for completing missing values. Y and X must have the same length, otherwise an error will be displayed. Input vectors do not contains NA, if not it a warning will be diplayed.

Author(s)

Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

Examples

```
data(dataFSMUMI)
X <- dataFSMUMI[, 1] ; Y <- dataFSMUMI[, 2]
compute.sim(Y,X)

# By definition, if true values is a constant vector
# and one or more imputed values are equal to the true values,
# similarity = 1.
X <- rep(5, 100)
Y <- X
compute.sim(Y,X)
```

`create.Myfis` *Building a fuzzy inference system*

Description

This function creates a fuzzy inference system that contains three input variables and three output variables. Each variable is expressed by 4 linguistic terms as low, medium, medium_high and high. A trapezoidal membership function is used to match input and output to a degree of membership.

Usage

```
create.Myfis()
```

Value

a fis object - a fuzzy inference system.

Author(s)

Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

`Creating_gaps` *Creating gaps in multivariate time series*

Description

This function builds gaps (with the same size) in multivariate time series. The size of each gap is defined as a percentage of input data length. By default, the position of each gap is randomly chosen.

Usage

```
Creating_gaps(data, rate, ngaps, begin = NULL)
```

Arguments

`data`	input multivariate time series
`rate`	size of a gap, as a percentage of input time series size
`ngaps`	number of gaps to create on each signal
`begin`	the starting position of a gap (random by default)

Value

Creating_gaps returns a multivariate time series with one gap per signal.

Author(s)

Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

Examples

```
data(dataFSMUMI)
X <- dataFSMUMI[1:5000,]; #reduction for demo
rate <- 0.1
ngaps <- 1
incompleted_signal <- Creating_gaps(X, rate,1)
```

Creating_gap_univariate

Creating a gap in a univariate series

Description

This function creates a gap (successive missing values) within a univariate signal. The size of gap is defined as a percentage of input vector length. By default, the starting position of the gap is chosen randomly.

Usage

```
Creating_gap_univariate(X, rate, begin = NULL)
```

Arguments

`X`	input vector
`rate`	size of desired gap, as a percentage of input vector size
`begin`	the begining position of the gap (random by default)

Value

This function returns a series with a gap of defined size.

Author(s)

Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

Examples

```
data(dataFSMUMI)
X <- dataFSMUMI[1:5000, 1] #reduction for demo
rate <- 0.1
incompleted_signal <- Creating_gap_univariate(X, rate)
```

`dataFSMUMI` *An example of a multivariate times series having three signals*

Description

An example of a multivariate times series having three signals

Usage

```
dataFSMUMI
```

Format

An object of class `matrix` with 32000 rows and 3 columns.

`evalfis` *FIS evaluation*

Description

Evaluate a Fuzzy Inference System (fis). For more details, please see FuzzyR package.

Usage

```
evalfis(input_stack, fis)
```

Arguments

`input_stack`	A matrix representing the input stack, number of inputs (columns) by number of outputs (rows).
`fis`	A fis must be provided.

Value

An evaluated crisp value for a given fis structure.

Author(s)

Jon Garibaldi, Chao Chen, Tajul Razak

FSMUMImputation — *Imputing large gaps based on the new fuzzy-weighted similarity measure*

Description

Fill large gaps in low or uncorrelated multivariate time series using the fuzzy-weighted similarity measure

Usage

```
FSMUMImputation(data, large_gap_threshold, step_threshold, step_finding)
```

Arguments

`data`	a multivariate signals containing gaps
`large_gap_threshold`	threshold used to determine a gap is large
`step_threshold`	increment used for finding the threshold
`step_finding`	increment used for retrieving a similar sequences to the queries

Value

returns a completed multivariate time series

Author(s)

Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

imp_1NA — *Completing isolated missing points (1NA)*

Description

Complete isolated missing points using the average of nearest neighbours.

Usage

```
imp_1NA(signal, pos1)
```

Arguments

`signal`	a univariate time series containing isolated missing values.
`pos1`	the position of the begining of gaps of size 1, obtained from Indexes_size_missing() function. This function returns a series with imputed values of all 1NAs.

Author(s)

Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

Indexes_size_missing *Indexes and sizes of gaps*

Description

Find the first positions of all gaps and their sizes respectively within a multivariate time series.

Usage

```
Indexes_size_missing(data)
```

Arguments

data	a multivariate data

Value

Return a list per signal in which each element containing the position of the starting of a gap (1st column) and its size (2nd column).

Author(s)

Thi-Thu-Hong Phan, Andre Bigand, Emilie Poisson-Caillault

Examples

```
data(dataFSMUMI)
X <- dataFSMUMI
rate <- 0.1
ngaps <- 1
incompleted_signal <- Creating_gaps(X, rate,1)
id_NA <- Indexes_size_missing(incompleted_signal)
```

Index