

Package ‘dsc’

September 27, 2023

Type Package

Title Dynamic Synthetic Control

Version 0.1.0

Date 2023-09-14

Description A package to implement the dynamic synthetic control method for time series analysis.

License MIT

Depends R ($\geq 3.5.0$)

Imports dplyr,

dtw,
forecast,
furry,
future,
ggplot2,
magrittr,
Matrix,
parallel,
purrr,
reshape2,
rlang,
signal,
stats,
Synth,
tibble,
zoo

Suggests testthat ($\geq 3.0.0$),

knitr,
rmarkdown

Encoding UTF-8

RoxygenNote 7.2.3

Config/testthat/edition 3

R topics documented:

add.buffer	2
dsc	2

minmax.normalize	5
normalize	5
preprocessing	6
RefTooShort	6
RemoveOutliers	7
t.normalize	7
warp2weight	8
warpWITHweight	8

Index	9
--------------	----------

add.buffer	<i>Add Buffer to Time Series</i>
------------	----------------------------------

Description

Adds a buffer to a time series using `auto.arima`.

Usage

```
add.buffer(TS, n)
```

Arguments

TS	Time series data.
n	Buffer size.

Value

Time series with added buffer.

dsc	<i>Dynamic Synthetic Control (DSC) method</i>
-----	---

Description

The DSC method combines ideas from the synthetic control method and dynamic time warping. This function processes the given data, applies time-series preprocessing, and then computes the synthetic control using dynamic time warping.

Usage

```
dsc(
  data,
  start.time,
  end.time,
  treat.time,
  dependent,
  k = 4,
  filter.width = 5,
```

```

buffer = 0,
norm.method = "t",
match.method = "fixed",
step.pattern1 = dtw::symmetricP1,
step.pattern2 = dtw::asymmetricP2,
plot.figures = FALSE,
n.burn = 3,
ma = 3,
ma.na = "original",
dist.quant = 1,
n.IQR = 3,
window.type = "none",
default.margin = 3,
n.q = 1,
n.r = 1,
parallel = TRUE,
rescale = TRUE,
dependent.id,
predictors,
special.predictors,
time.predictors.prior,
time.optimize.ssr
)

```

Arguments

<code>data</code>	A data frame containing the observational data.
<code>start.time</code>	Starting time for the analysis.
<code>end.time</code>	Ending time for the analysis.
<code>treat.time</code>	Treatment time.
<code>dependent</code>	The dependent variable name in the dataset.
<code>k</code>	Integer, number of control units used for dynamic time warping. Default is 4.
<code>filter.width</code>	Integer, width of the filter. Default is 5.
<code>buffer</code>	Integer, buffer for time series alignment. Default is 0.
<code>norm.method</code>	Method for normalization. Default is "t".
<code>match.method</code>	Method for matching. Default is "fixed".
<code>step.pattern1</code>	Step pattern for the DTW. Default is <code>dtw::symmetricP1</code> .
<code>step.pattern2</code>	Alternative step pattern for DTW. Default is <code>dtw::asymmetricP2</code> .
<code>plot.figures</code>	Logical, if TRUE plots will be generated. Default is FALSE.
<code>n.burn</code>	Integer, number of initial time periods to disregard. Default is 3.
<code>ma</code>	Integer, moving average length. Default is 3.
<code>ma.na</code>	Method to handle missing values in moving average. Default is "original".
<code>dist.quant</code>	Numeric, quantile for distance measure. Default is 1.
<code>n.IQR</code>	Numeric, factor for IQR in outlier detection. Default is 3.
<code>window.type</code>	Type of window for DTW. Default is "none".
<code>default.margin</code>	Default margin size. Default is 3.
<code>n.q</code>	Integer, number of synthetic controls to use. Default is 1.

<code>n.r</code>	Integer, number of predictors to use. Default is 1.
<code>parallel</code>	Logical, if TRUE parallel processing will be enabled. Default is TRUE.
<code>rescale</code>	Logical, if TRUE data will be rescaled. Default is TRUE.
<code>dependent.id</code>	Numeric, ID of the dependent unit.
<code>predictors</code>	List, names of predictor variables.
<code>special.predictors</code>	List, names of special predictor variables.
<code>time.predictors.prior</code>	List, names of time predictor variables for the prior period.
<code>time.optimize.ssr</code>	List, names of time predictor variables for the SSR optimization.

Value

A list containing results of the synthetic control analysis.

Examples

```
## Not run:
library(dsc)

# Load the Basque dataset from the Synth package
data(basque, package = "Synth")
data <- basque

# Rename relevant columns for clarity
colnames(data)[1:4] <- c("id", "unit", "time", "value")

# Compute additional variables
data$invest_ratio <- data$invest / data$value
data$value_raw <- data$value

# Define special predictors for the model
special_preds <- expression(list(
  list(dep.var, 1960:1969, c("mean")),
  list("invest_ratio", 1964:1969, c("mean")),
  list("popdens", 1969, c("mean")),
  list("sec.agriculture", 1961:1969, c("mean")),
  list("sec.energy", 1961:1969, c("mean")),
  list("sec.industry", 1961:1969, c("mean")),
  list("sec.construction", 1961:1969, c("mean")),
  list("sec.services.venta", 1961:1969, c("mean")),
  list("sec.services.nonventa", 1961:1969, c("mean")),
  list("school.illit", 1964:1969, c("mean")),
  list("school.prim", 1964:1969, c("mean")),
  list("school.med", 1964:1969, c("mean")),
  list("school.high", 1964:1969, c("mean")),
  list("school.post.high", 1964:1969, c("mean"))
))

# Execute the DSC analysis
result <- dsc(
  data = data,
  start.time = 1955,
```

```

    end.time = 1997,
    treat.time = 1970,
    dependent = "Basque Country (Pais Vasco)",
    predictors = NULL,
    parallel = TRUE,
    special.predictors = special_preds,
    time.predictors.prior = 1955:1969,
    time.optimize.ssr = 1955:1969
)

## End(Not run)

```

minmax.normalize	<i>Min-Max Normalization</i>
------------------	------------------------------

Description

Normalizes the data using min-max scaling.

Usage

```
minmax.normalize(data, reference = NULL)
```

Arguments

data	A numeric vector to be normalized.
reference	An optional reference for normalization.

Value

Normalized data.

normalize	<i>General Normalization Function</i>
-----------	---------------------------------------

Description

Applies the specified normalization to the data.

Usage

```
normalize(data, norm.method, reference = NULL)
```

Arguments

data	A numeric vector to be normalized.
norm.method	A string that specifies the normalization method.
reference	An optional reference for normalization.

Value

Normalized data.

preprocessing	<i>Pre-process Data</i>
---------------	-------------------------

Description

Applies several pre-processing steps to the data.

Usage

```
preprocessing(
  data,
  filter.width = 5,
  norm.method = "t",
  n.poly = 3,
  n.deriv = 2,
  plot.data = FALSE
)
```

Arguments

<code>data</code>	The data to be processed.
<code>filter.width</code>	Width of the filter.
<code>norm.method</code>	Normalization method.
<code>n.poly</code>	Degree of the polynomial.
<code>n.deriv</code>	Order of the derivative.
<code>plot.data</code>	Logical, indicating whether to plot the data.

Value

Processed data.

RefTooShort	<i>Check if Reference is Short in DTW</i>
-------------	---

Description

Checks if the reference series is too short for dynamic time warping.

Usage

```
RefTooShort(
  query,
  reference,
  step.pattern = dtw::symmetricP2,
  window.type = "none",
  window.size = NULL
)
```

Arguments

query	The query series.
reference	The reference series.
step.pattern	Step pattern for DTW.
window.type	Type of window for DTW.
window.size	Size of the window for DTW.

Value

Logical indicating if reference is too short.

RemoveOutliers	<i>Remove Outliers in Weight Matrix</i>
----------------	---

Description

Removes outliers from the data based on interquartile range.

Usage

```
RemoveOutliers(data, n.IQR = 3)
```

Arguments

data	Numeric data.
n.IQR	Multiplier for the interquartile range.

Value

Data with outliers removed.

t.normalize	<i>Standard Z-score Normalization</i>
-------------	---------------------------------------

Description

Normalizes the data using z-score.

Usage

```
## S3 method for class 'normalize'  
t(data, reference = NULL)
```

Arguments

data	A numeric vector to be normalized.
reference	An optional reference for normalization.

Value

Normalized data.

`warp2weight`*Convert Warping Path to Weight*

Description

Transforms a warping path to weights.

Usage

```
warp2weight(W)
```

Arguments

`W` The warping path.

Value

Weights.

`warpWITHweight`*Warp Time Series with Weights*

Description

Warpes a time series using the provided weights.

Usage

```
warpWITHweight(ts, weight)
```

Arguments

`ts` Time series to be warped.
`weight` Weights for warping.

Value

Warped time series.

Index

`add.buffer`, [2](#)

`dsc`, [2](#)

`minmax.normalize`, [5](#)

`normalize`, [5](#)

`preprocessing`, [6](#)

`RefTooShort`, [6](#)

`RemoveOutliers`, [7](#)

`t.normalize`, [7](#)

`warp2weight`, [8](#)

`warpWITHweight`, [8](#)