Package 'dsc'

September 27, 2023

Type Package

Title Dynamic Synthetic Control

Version 0.1.0

Date 2023-09-14

Description Implements the Dynamic Synthetic Control method as described in Cao and Chadefaux (2023). Synthetic controls are widely used in social science research to estimate the causal effects of treatments such as events or policies. The 'dsc' package extends traditional synthetic control methods by introducing a dynamic approach that accounts for varying speeds at which different units respond to changes. Ignoring these varying speeds can result in biased estimates of causal effects. This package offers a more robust method for constructing counterfactuals in time series analysis, thereby improving the accuracy of treatment effect estimates. It incorporates algorithms for dynamic time warping and allows for adjustment of varying speeds within and across units. The package is validated through extensive Monte-Carlo simulations and applied to re-estimate the effects of several seminal case studies.

License MIT

URL github.com/conflictlab/dsc

Depends R (>= 3.5.0)

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

Suggests testthat (>= 3.0.0),
knitr,
rmarkdown

Encoding UTF-8

RoxygenNote 7.2.3

Config/testthat/edition 3

R topics documented:

`add.buffer` *Add Buffer to Time Series*

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 *Dynamic Synthetic Control (DSC) method*

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

data	A data frame containing the observational data.
start.time	Starting time for the analysis.
end.time	Ending time for the analysis.
treat.time	Treatment time.
dependent	The dependent variable name in the dataset.

k	Integer, number of control units used for dynamic time warping. Default is 4.
filter.width	Integer, width of the filter. Default is 5.
buffer	Integer, buffer for time series alignment. Default is 0.
norm.method	Method for normalization. Default is "t".
match.method	Method for matching. Default is "fixed".
step.pattern1	Step pattern for the DTW. Default is dtw::symmetricP1.
step.pattern2	Alternative step pattern for DTW. Default is dtw::asymmetricP2.
plot.figures	Logical, if TRUE plots will be generated. Default is FALSE.
n.burn	Integer, number of initial time periods to disregard. Default is 3.
ma	Integer, moving average length. Default is 3.
ma.na	Method to handle missing values in moving average. Default is "original".
dist.quant	Numeric, quantile for distance measure. Default is 1.
n.IQR	Numeric, factor for IQR in outlier detection. Default is 3.
window.type	Type of window for DTW. Default is "none".
default.margin	Default margin size. Default is 3.
n.q	Integer, number of synthetic controls to use. Default is 1.
n.r	Integer, number of predictors to use. Default is 1.
parallel	Logical, if TRUE parallel processing will be enabled. Default is TRUE.
rescale	Logical, if TRUE data will be rescaled. Default is TRUE.
dependent.id	Numeric, ID of the dependent unit.
predictors	List, names of predictor variables.
special.predictors	List, names of special predictor variables.
time.predictors.prior	List, names of time predictor variables for the prior period.
time.optimize.ssr	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` *Min-Max Normalization*

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 *General Normalization Function*

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 *Pre-process Data*

Description

Applies several pre-processing steps to the data.

Usage

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

Arguments

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

Value

Processed data.

RefTooShort *Check if Reference is Short in DTW*

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 *Remove Outliers in Weight Matrix*

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 *Standard Z-score Normalization*

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

Warps 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