Package 'tsrobprep'

April 11, 2021

Title Robust Preprocessing of Time Series Data

Version 0.1.0

Date 2021-04-11

Description Methods for handling the missing values outliers are introduced in this package. The recognized missing values and outliers are replaced using a model-based approach. The model may consist of both autoregressive components and external regressors. The methods work robust and efficient, and they are fully tunable. The primary motivation for writing the package was preprocessing of the energy systems data, e.g. power plant production time series, but the package could be used with any time series data.

Depends R (>= 3.2.0)

License MIT + file LICENSE

Encoding UTF-8

Imports Matrix, quantreg, mclust, MASS

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Author Michał Narajewski [aut, cre] (<https://orcid.org/0000-0002-3115-0162>), Florian Ziel [aut] (<https://orcid.org/0000-0002-2974-2660>), Jens Kley-Holsteg [ctb]

Maintainer Michał Narajewski <michal.narajewski@uni-due.de>

Repository CRAN

Date/Publication 2021-04-11 14:00:02 UTC

R topics documented:

`auto_data_cleaning` *Perform automatic data cleaning of time series data*

Description

Returns a matrix or a list of matrices with imputed missing values and outliers. The function automatizes the usage of functions model_missing_data, detect_outliers and impute_modelled_data. The function is designed for numerical data only.

Usage

```
auto_data_cleaning(
  data,
  S,
  tau = 0.5,
  no.of.last.indices.to.fix = S[1],
  indices.to.fix = NULL,
  model.missing.pars = list(),
  detect.outliers.pars = list()
)
```

Arguments

`data`	an input vector, matrix or data frame of dimension nobs x nvars containing missing values; each column is a variable.
`S`	a number or vector describing the seasonalities (S_1, ..., S_K) in the data, e.g. c(24, 168) if the data consists of 24 observations per day and there is a weekly seasonality in the data.
`tau`	the quantile(s) of the missing values to be estimated in the quantile regression. Tau accepts all values in (0,1), the default is 0.5.
`no.of.last.indices.to.fix`	a number of observations in the tail of the data to be fixed, by default set to S.
`indices.to.fix`	indices of the data to be fixed. If NULL, then it is calculated based on the no.of.last.indices.to.fix parameter. Otherwise, the no.of.last.indices.to.fix parameter is ignored.
`model.missing.pars`	named list containing additional arguments for the model_missing_data function.
`detect.outliers.pars`	named list containing additional arguments for the detect_outliers function.

Details

The function calls model_missing_data to clean the data from missing values, detect_outliers to detect outliers, removes them and finally applies again model_missing_data function. For details see the functions' respective help sections.

Value

A list which contains a matrix or a list of matrices with imputed missing values or outliers, the indices of the data that were modelled, and the given quantile values.

See Also

model_missing_data, detect_outliers, impute_modelled_data

Examples

```
## Not run:
id <- 14000:17000
autoclean <- auto_data_cleaning(data = GBload[id,-1], S = c(48, 7*48), tau = 0.5,
             no.of.last.indices.to.fix = length(id),
             model.missing.pars = list(consider.as.missing = 0, min.val = 0),
            detect.outliers.pars = list(model.select.iter = 1, outlier.detect.iter = 1))
autoclean$replaced.indices

## End(Not run)
```

`detect_outliers` *Detects unreliable outliers in univariate time series data based on finite mixture modelling*

Description

This function applies finite mixture modelling to compute the probability of each observation being an outlier in an univariate time series. By utilizing the Mclust package the data is assigned in k clusters whereof one contains outliers. The clustering process is based on features, which are specifically desinged for outlier detection in time series data.

Usage

```
detect_outliers(
  data,
  S,
  model.select.iter = 10,
  outlier.detect.iter = 10,
  proba = 0.1,
  share = 0.3,
  detection.parameter = 1,
  out.par = 2,
  noise.par = 10^-5,
  mar = 1,
  max.cluster = 9,
  G = NULL,
  modelName = NULL,
 feat.int = list(org.s = TRUE, grad = TRUE, abs.grad = TRUE, abs.seas.grad = TRUE,
```

```
    lin.tr = TRUE),
  ...
)
```

Arguments

data	an one dimensional matrix or data frame without missing data; each row is an observation.
S	vector with numeric values for each seasonality present in data.
model.select.iter	denotes the number of iterations to find the optimal number of clusters as well as the optimal model of the covariance matrix. By default set to 10. Results become more robust with increasing iterations but lead likewise to increasing computational time.
outlier.detect.iter	denotes the number of iterations for outlier detection based on variing subsamples controlled by the share parameter. By default set to 10. Results become more robust with increasing iterations but lead likewise to increasing computational time.
proba	denotes threshold ranging from 0 to 1 from which an observation is considered being an outlier. Number of outlier increases with decreasing threshold. By default set to 0.1, implying that in average an observation obtains a probability in each iteration of 0.1 belonging to the outlier cluster.
share	controlls the size of the subsample used for estimation. By default set to 0.3 (ranging from 0 to 1). Controlls the computational time and the robustness of the method.
detection.parameter	denotes a parameter to regulate the detection sensisitivity. By default set to 1. The smaller the more outliers.
out.par	controls the number of artifially produced outliers in relation to the subsample size controlled by the share parameter. By default out.par ist set to 2. By increase a priori it is assumed that share of outliers in data increases.
noise.par	controls strenght of noise added to feature matrix to avoid zero variance issues by applying bivariate features. By default set to 10^-5, strength of noise decreases with decreasing noise.par parameter.
mar	denotes a margin controlling the number of adjacent values around an identified outlier which are likewise considered as outliers. By default set to 1 so that the one most closest neighbours of an identified outlier on each side are also treated as outliers.
max.cluster	a single numeric value controlling the maximum number of clusters allowed. By default set to 9.
G	denotes the optimal number of clusters limited by the max.cluster paramter. By default G is set to NULL and automatically calculated based on the BIC.
modelName	denotes the geometric features of the covariance matrix. i.e. "EII", "VII", "EEI", "EVI", "VEI", "VVI", etc.. By default modelName is set to NUll and automatically calcualted based on BIC. The help file for mclustModelNames describes the available models.

`feat.int` a list of logical values indicating which features should be applied in clustering algorithm.

org.s denotes the scaled original time series.

lin.tr denotes linear trends based on seasonalities S.

grad denotes the gradient of scaled original time series.

abs.grad denotes absolute gradient of scaled original time series.

abs.seas.grad denotes the absolute seasonal gradient of scaled original time series based on seasonalties S.

`...` additional arguments for the Mclust function.

Details

The detection of outliers is addressed by model based clustering based on parameterized finite Gaussian mixture models. For cluster estimation the Mclust function is applied. Models are estimated by the EM algorithm initialized by hierarchical model-based agglomerative clustering. The optimal model can be selected according to BIC.

Value

An object of class "tsrobprep" containing the following elements:

original.data an numeric vector containing the original data.

outlier.probs an numeric vector containing the averaged probability

outlier.probs.mat a matrix containg the probability for each iteration that observation is belonging to the outlier cluster. Each row is an observation and each column an iteration.

outlier.pos a logical vector indicating the position of each outlier.

outlier.pos.aug a logical vector indicating the position of each outlier including neigbouring values based on the mar parameter.

estimated.models a list containing each estimated model.

BIC an mclustBIC object containing the Bayesian Information Criterion for the specified mixture models numbers of clusters. Auxiliary information returned as attributes.

See Also

model_missing_data, impute_modelled_data, auto_data_cleaning

Examples

```
## Not run:
set.seed(1)
id <- 14000:17000
# Replace missing values
modelmd <- model_missing_data(data = GBload[id, -1], tau = 0.5,
 S = c(48, 336), indices.to.fix = seq_len(nrow(GBload[id, ])),
 consider.as.missing = 0, min.val = 0)
# Impute missing values
 data.imputed <- impute_modelled_data(modelmd)
```

```
# Detect outliers
o.ident <- detect_outliers(data = data.imputed, S = c(48, 336),
                           model.select.iter = 1,
                           outlier.detect.iter = 1)
# Plot of identified outliers in time series
plot(data.imputed, type = "o", col=1 + o.ident$outlier.pos.aug,
     pch = 1 + 18 * o.ident$outlier.probs)

# Plot of feature matrix
plot.ts(o.ident$features, type = "o",
        col = 1 + o.ident$outlier.pos,
        pch = 1 + 18 * o.ident$outlier.probs)

## End(Not run)
```

GBload *The electricity actual total load in Great Britain in year 2018*

Description

A dataset containing the electricity actual total load (MW) in Great Britain in year 2018 presented in half-hour interval. Each data point regards 30 minutes of electricity load starting at given time. The data consists of both missing values and outliers.

Usage

```
GBload
```

Format

A data frame with 17520 rows and 2 variables:

Date date indicating the delivery beginning of the electricity

Load actual electricity load in MW ...

Source

https://transparency.entsoe.eu/

impute_modelled_data *Impute modelled missing time series data*

Description

Returns a matrix or a list of matrices with imputed missing values or outliers. As argument the function requires an object of class "tsrobprep" and the quantiles to be imputed.

Usage

```
impute_modelled_data(object, tau = NULL)
```

Arguments

object	an object of class "tsrobprep" that is an output of function model_missing_data.
tau	the quantile(s) of the missing values to be imputed. tau should be a subset of the quantile values present in the "tsrobprep" object. By default all quantiles present in the object are used.

Value

A matrix or a list of matrices with imputed missing values or outliers.

See Also

model_missing_data, detect_outliers, auto_data_cleaning

Examples

```
model.miss <- model_missing_data(data = GBload[,-1], S = c(48,7*48), tau = 0.5,
              no.of.last.indices.to.fix = dim(GBload)[1], consider.as.missing = 0,
              min.val = 0)
model.miss$estimated.models
model.miss$replaced.indices
new.GBload <- impute_modelled_data(model.miss)
```

model_missing_data *Model missing time series data*

Description

Returns an object of class "tsrobprep" which contains the original data and the modelled missing values to be imputed. The function model_missing_data models missing values in a time series data using the quantile regression implemented in quantreg package. The model uses autoregression on the time series as explanatory variables as well as the provided external variables. The function is designed for numerical data only.

Usage

```
model_missing_data(
  data,
  S,
  tau = 0.5,
  no.of.last.indices.to.fix = S[1],
  indices.to.fix = NULL,
  p = NULL,
  mirror = FALSE,
  lags = NULL,
  extreg = NULL,
  n.best.extreg = NULL,
  use.data.as.ext = FALSE,
  lag.externals = FALSE,
  consider.as.missing = NULL,
  whole.period.missing.only = FALSE,
  min.val = -Inf,
  max.val = Inf,
  Cor_thres = 0.5,
  digits = 3,
  ...
)
```

Arguments

data	an input vector, matrix or data frame of dimension nobs x nvars containing missing values; each column is a variable.
S	a number or vector describing the seasonalities (S_1, ..., S_K) in the data, e.g. c(24, 168) if the data consists of 24 observations per day and there is a weekly seasonality in the data.
tau	the quantile(s) of the missing values to be estimated in the quantile regression. Tau accepts all values in (0,1), the default is 0.5.
no.of.last.indices.to.fix	a number of observations in the tail of the data to be fixed, by default set to S.
indices.to.fix	indices of the data to be fixed. If NULL, then it is calculated based on the no.of.last.indices.to.fix parameter. Otherwise, the no.of.last.indices.to.fix parameter is ignored.
p	a number or vector of length(S) = K indicating the order of a K-seasonal autoregressive process to be estimated. If NULL, chosen data-based.
mirror	if TRUE then autoregressive lags up to order p are not only added to the seasonalities but also subtracted.
lags	a numeric vector with the lags to use in the autoregression. Negative values are accepted and then also the "future" observations are used for modelling. If not NULL, p and mirror are ignored.
extreg	a vector, matrix or data frame of data containing external regressors; each column is a variable.

`n.best.extreg` a numeric value specifying the maximal number of considered best correlated external regressors (selected in decreasing order). If NULL, then all variables in extreg are used for modelling.

`use.data.as.ext` logical specifying whether to use the remaining variables in the data as external regressors or not.

`lag.externals` logical specifying whether to lag the external regressors or not. If TRUE, then the algorithm uses the lags specified in parameter lags.

`consider.as.missing` a vector of numerical values which are considered as missing in the data.

`whole.period.missing.only` if FALSE, then all observations which correspond to the values of consider.as.missing are treated as missings. If TRUE, then only consecutive observations of specified length are considered (length is defined by S).

`min.val` a single value or a vector of length nvars providing the minimum possible value of each variable in the data. If a single value, then it applies to all variables. By default set to -Inf.

`max.val` a single value or a vector of length nvars providing the maximum possible value of each variable in the data. If a single value, then it applies to all variables. By default set to Inf.

`Cor_thres` a single value providing the correlation threshold from which external regressors are considered in the quantile regression.

`digits` integer indicating the number of decimal places allowed in the data, by default set to 3.

`...` additional arguments for the rq.fit.fnb algorithm.

Details

The function uses quantile regression in order to model missing values and prepare it for imputation. In this purpose the rq.fit.fnb function from quantreg package is used. The function computes the quantile regression methods utilizing the Frisch-Newton algorithm for user-specified quantile values. The modelled values can be imputed using impute_modelled_data function.

Value

An object of class "tsrobprep" which contains the original data, the indices of the data that were modelled, the given quantile values, a list of sparse matrices with the modelled data to be imputed and a list of the numbers of models estimated for every variable.

See Also

impute_modelled_data, detect_outliers, auto_data_cleaning

Examples

```
model.miss <- model_missing_data(data = GBload[,-1], S = c(48,7*48), tau = 0.5,
              no.of.last.indices.to.fix = dim(GBload)[1], consider.as.missing = 0,
              min.val = 0)
model.miss$estimated.models
model.miss$replaced.indices
new.GBload <- impute_modelled_data(model.miss)
```

Index