Package ‘theft’

March 27, 2023

Type Package

Title Tools for Handling Extraction of Features from Time Series

Version 0.4.2.4

Date 2023-03-27

Maintainer Trent Henderson <then6675@uni.sydney.edu.au>


BugReports https://github.com/hendersontrent/theft/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports rlang, stats, dplyr, ggplot2, tidyr, reshape2, scales, tibble, purrr, broom, tsibble, fabletools, tsfeatures, feasts, Rcatch22, reticulate, Rtsne, R.matlab, caret, janitor

Suggests lifecycle, cachem, bslib, knitr, markdown, rmarkdown, pkgdown, testthat

RoxygenNote 7.2.2
calculate_features

Compute features on an input time series dataset

Description

Compute features on an input time series dataset

Usage

```r
calculate_features(  
  data,  
  id_var = "id",  
  time_var = "timepoint",  
  values_var = "values",  
)```
calculate_features

group_var = NULL,
feature_set = c("catch22", "feasts", "tsfeatures", "Kats", "tsfresh", "TSFEL"),
catch24 = FALSE,
tsfresh_cleanup = FALSE,
seed = 123
)

Arguments

data data.frame with at least 4 columns: id variable, group variable, time variable, value variable
id_var string specifying the ID variable to identify each time series. Defaults to "id"
time_var string specifying the time index variable. Defaults to "timepoint"
values_var string specifying the values variable. Defaults to "values"
group_var string specifying the grouping variable that each unique series sits under (if one exists). Defaults to NULL
feature_set string or vector of strings denoting the set of time-series features to calculate. Defaults to "catch22"
catch24 Boolean specifying whether to compute catch24 in addition to catch22 if catch22 is one of the feature sets selected. Defaults to FALSE
tsfresh_cleanup Boolean specifying whether to use the in-built tsfresh relevant feature filter or not. Defaults to FALSE
seed integer denoting a fixed number for R’s random number generator to ensure reproducibility

Value

object of class feature_calculations that contains the summary statistics for each feature

Author(s)

Trent Henderson

Examples

featMat <- calculate_features(data = simData,
id_var = "id",
time_var = "timepoint",
values_var = "values",
group_var = "process",
feature_set = "catch22",
seed = 123)
check_vector_quality  Check for presence of NAs and non-numerics in a vector

Description
Check for presence of NAs and non-numerics in a vector

Usage
check_vector_quality(x)

Arguments
x  input vector

Value
Boolean of whether the data is good to extract features on or not

Author(s)
Trent Henderson

compute_top_features  Return an object containing results from top-performing features on a classification task

Description
Return an object containing results from top-performing features on a classification task

Usage
compute_top_features(
  data,
  num_features = 40,
  normalise_violin_plots = FALSE,
  method = c("z-score", "Sigmoid", "RobustSigmoid", "MinMax"),
  cor_method = c("pearson", "spearman"),
  test_method = "gaussprRadial",
  clust_method = c("average", "ward.D", "ward.D2", "single", "complete", "mcquitty", "median", "centroid"),
  use_balanced_accuracy = FALSE,
  use_k_fold = FALSE,
  num_folds = 10,
  use_empirical_null = FALSE,
null_testing_method = c("ModelFreeShuffles", "NullModelFits"),
p_value_method = c("empirical", "gaussian"),
num_permutations = 50,
pool_empirical_null = FALSE,
seed = 123
)

Arguments

data the feature_calculations object containing the raw feature matrix produced by calculate_features
num_features integer denoting the number of top features to retain and explore. Defaults to 40
normalise_violin_plots Boolean of whether to normalise features before plotting. Defaults to FALSE
method a rescaling/normalising method to apply to violin plots. Defaults to "z-score"
cor_method string denoting the correlation method to use. Defaults to "pearson"
test_method string specifying the algorithm to use for quantifying class separation. Defaults to "gaussprRadial". Should be either "t-test", "wilcox", or "binomial logistic" for two-class problems to obtain exact statistics, or a valid caret classification model for everything else
clust_method string denoting the hierarchical clustering method to use for the pairwise correlation plot. Defaults to "average"
use_balanced_accuracy Boolean specifying whether to use balanced accuracy as the summary metric for caret model training. Defaults to FALSE
use_k_fold Boolean specifying whether to use k-fold procedures for generating a distribution of classification accuracy estimates if a caret model is specified for test_method. Defaults to FALSE
num_folds integer specifying the number of k-folds to perform if use_k_fold is set to TRUE. Defaults to 10
use_empirical_null Boolean specifying whether to use empirical null procedures to compute p-values if a caret model is specified for test_method. Defaults to FALSE
null_testing_method string specifying the type of statistical method to use to calculate p-values. Defaults to "ModelFreeShuffles"
p_value_method string specifying the method of calculating p-values. Defaults to "empirical"
num_permutations integer specifying the number of class label shuffles to perform if use_empirical_null is TRUE. Defaults to 50
pool_empirical_null Boolean specifying whether to use the pooled empirical null distribution of all features or each features’ individual empirical null distribution if a caret model is specified for test_method use_empirical_null is TRUE. Defaults to FALSE
seed integer denoting a fixed number for R’s random number generator to ensure reproducibility
**feature_list**

- **Value**
  - an object of class `list` containing a `data.frame` of results, a `ggplot` feature x feature matrix plot, and a `ggplot` violin plot

- **Author(s)**
  - Trent Henderson

- **Examples**

```r
featMat <- calculate_features(data = simData,
   id_var = "id",
   time_var = "timepoint",
   values_var = "values",
   group_var = "process",
   feature_set = "catch22",
   seed = 123)

compute_top_features(featMat,
   num_features = 10,
   normalise_violin_plots = FALSE,
   method = "RobustSigmoid",
   cor_method = "pearson",
   test_method = "gaussprRadial",
   clust_method = "average",
   use_balanced_accuracy = FALSE,
   use_k_fold = FALSE,
   num_folds = 10,
   use_empirical_null = TRUE,
   null_testing_method = "ModelFreeShuffles",
   p_value_method = "gaussian",
   num_permutations = 100,
   pool_empirical_null = FALSE,
   seed = 123)
```

---

**feature_list**

All features available in theft in tidy format

**Description**

The variables include:

**Usage**

`feature_list`
fit_multi_feature_classifier

Format
A tidy data frame with 2 variables:

feature_set Name of the set the feature is from
feature Name of the feature

fit_multi_feature_classifier
Fit a classifier to feature matrix using all features or all features by set

Description
Fit a classifier to feature matrix using all features or all features by set

Usage

fit_multi_feature_classifier(
  data,
  by_set = FALSE,
  test_method = "gaussprRadial",
  use_balanced_accuracy = FALSE,
  use_k_fold = TRUE,
  num_folds = 10,
  use_empirical_null = FALSE,
  null_testing_method = c("ModelFreeShuffles", "NullModelFits"),
  p_value_method = c("empirical", "gaussian"),
  num_permutations = 100,
  seed = 123
)

Arguments

data the feature_calculations object containing the raw feature matrix produced
by calculate_features

by_set Boolean specifying whether to compute classifiers for each feature set. Defaults
to FALSE

test_method string specifying the algorithm to use for quantifying class separation. De-
defaults to "gaussprRadial". Must be a valid caret classification model

use_balanced_accuracy Boolean specifying whether to use balanced accuracy as the summary metric
for caret model training. Defaults to FALSE

use_k_fold Boolean specifying whether to use k-fold procedures for generating a distribu-
tion of classification accuracy estimates. Defaults to TRUE

num_folds integer specifying the number of folds (train-test splits) to perform if use_k_fold
is set to TRUE. Defaults to 10
use_empirical_null  
Boolean specifying whether to use empirical null procedures to compute p-values. Defaults to FALSE

null_testing_method  
string specifying the type of statistical method to use to calculate p-values. Defaults to model free shuffles

p_value_method  
string specifying the method of calculating p-values. Defaults to "empirical"

num_permutations  
integer specifying the number of class label shuffles to perform if use_empirical_null is TRUE. Defaults to 100

seed  
integer denoting a fixed number for R’s random number generator to ensure reproducibility

Value
an object of class list containing a data.frame summary of raw classification results, a data.frame summary of the test statistics, and a ggplot object if by_set is TRUE

Author(s)
Trent Henderson

Examples

featMat <- calculate_features(data = simData,  
id_var = "id",  
time_var = "timepoint",  
values_var = "values",  
group_var = "process",  
feature_set = "catch22",  
seed = 123)

fit_multi_feature_classifier(featMat,   
by_set = FALSE,  
test_method = "gaussprRadial",  
use_balanced_accuracy = FALSE,  
use_k_fold = TRUE,  
num_folds = 10,  
use_empirical_null = TRUE,  
null_testing_method = "ModelFreeShuffles",  
p_value_method = "gaussian",  
num_permutations = 50,  
seed = 123)
fit_single_feature_classifier

Fit a classifier to feature matrix to extract top performers

Description

Fit a classifier to feature matrix to extract top performers

Usage

fit_single_feature_classifier(
  data,
  test_method = "gaussprRadial",
  use_balanced_accuracy = FALSE,
  use_k_fold = FALSE,
  num_folds = 10,
  use_empirical_null = FALSE,
  null_testing_method = c("ModelFreeShuffles", "NullModelFits"),
  p_value_method = c("empirical", "gaussian"),
  num_permutations = 50,
  pool_empirical_null = FALSE,
  seed = 123
)

Arguments

data the data.frame containing the raw feature matrix

test_method string specifying the algorithm to use for quantifying class separation. Defaults to "gaussprRadial". Should be either "t-test", "wilcox", or "binomial logistic" for two-class problems to obtain exact statistics, or a valid caret classification model for everything else

use_balanced_accuracy Boolean specifying whether to use balanced accuracy as the summary metric for caret model training. Defaults to FALSE

use_k_fold Boolean specifying whether to use k-fold procedures for generating a distribution of classification accuracy estimates if a caret model is specified for test_method. Defaults to FALSE

num_folds integer specifying the number of k-folds to perform if use_k_fold is set to TRUE. Defaults to 10

use_empirical_null Boolean specifying whether to use empirical null procedures to compute p-values if a caret model is specified for test_method. Defaults to FALSE

null_testing_method string specifying the type of statistical method to use to calculate p-values. Defaults to model free shuffles
p_value_method string specifying the method of calculating p-values. Defaults to "empirical"
num_permutations integer specifying the number of class label shuffles to perform if use_empirical_null is TRUE. Defaults to 50
pool_empirical_null Boolean specifying whether to use the pooled empirical null distribution of all features or each features' individual empirical null distribution if a caret model is specified for test_method use_empirical_null is TRUE. Defaults to FALSE
seed integer denoting a fixed number for R's random number generator to ensure reproducibility

Value
an object of class data.frame

Author(s)
Trent Henderson

init_theft(python_path, venv_path)

Arguments
python_path string specifying the filepath to the version of Python you wish to use
venv_path string specifying the filepath to the Python virtual environment where "tsfresh", "tsfel", and/or "kats" are installed

Value
no return value; called for side effects

Author(s)
Trent Henderson
**minmax_scaler**

Rescales a numeric vector into the unit interval [0,1]

### Description

\[ z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \]

### Usage

`minmax_scaler(x)`

#### Arguments

- `x` numeric vector

#### Value

numeric vector

#### Author(s)

Trent Henderson

---

**normalise**

Scale each feature vector into a user-specified range for visualisation and modelling

### Description

Scale each feature vector into a user-specified range for visualisation and modelling

### Usage

`normalise(data, method = c("z-score", "Sigmoid", "RobustSigmoid", "MinMax"))`

#### Arguments

- `data` either a `feature_calculations` object containing the raw feature matrix produced by `calculate_features` or a vector of class `numeric` containing values to be normalised
- `method` string denoting the rescaling/normalising method to apply to violin plots. Defaults to "z-score"

#### Value

either an object of class `data.frame` or `numeric`
plot.feature_calculations

**Author(s)**

Trent Henderson

---

**Description**

Produce a plot for a feature_calculations object

**Usage**

```r
## S3 method for class 'feature_calculations'
plot(
  x,
  type = c("quality", "matrix", "cor"),
  method = c("z-score", "Sigmoid", "RobustSigmoid", "MinMax"),
  clust_method = c("average", "ward.D", "ward.D2", "single", "complete", "mcquitty", "median", "centroid"),
  cor_method = c("pearson", "spearman"),
  ...
)
```

**Arguments**

- `x` the feature_calculations object containing the raw feature matrix produced by `calculate_features`
- `type` string specifying the type of plot to draw. Defaults to "quality"
- `method` string specifying a rescaling/normalising method to apply if `type = "matrix"` or if `type = "cor"`. Defaults to "z-score"
- `clust_method` string specifying the hierarchical clustering method to use if `type = "matrix"` or if `type = "cor"`. Defaults to "average"
- `cor_method` string specifying the correlation method to use if `type = "cor"`. Defaults to "pearson"
- `...` Arguments to be passed to methods

**Value**

object of class ggplot that contains the graphic

**Author(s)**

Trent Henderson
### plot.low_dimension

*Produce a plot for a low_dimension object*

#### Description

Produce a plot for a low_dimension object

#### Usage

```r
# S3 method for class 'low_dimension'
plot(x, show_covariance = TRUE, ...)
```

#### Arguments

- `x`: the low_dimension object containing the dimensionality reduction projection calculated by reduce_dims
- `show_covariance`: Boolean of whether covariance ellipses should be shown on the plot. Defaults to TRUE
- `...`: Arguments to be passed to methods

#### Value

object of class ggplot that contains the graphic

#### Author(s)

Trent Henderson

### process_hctsa_file

*Load in hctsa formatted MATLAB files of time series data into a tidy format ready for feature extraction*

#### Description

Load in hctsa formatted MATLAB files of time series data into a tidy format ready for feature extraction

#### Usage

```r
process_hctsa_file(data)
```

#### Arguments

- `data`: string specifying the filepath to the MATLAB file to parse
**reduce_dims**

Value
an object of class `data.frame` in tidy format

Author(s)
Trent Henderson

Examples

```r
myfile <- process_hctsa_file("https://cloudstor.aarnet.edu.au/plus/s/6sRD6IPMjyZLN1N/download")
```

**Description**
Project a feature matrix into a low dimensional representation using PCA or t-SNE

**Usage**
```r
reduce_dims(
  data,
  method = c("z-score", "Sigmoid", "RobustSigmoid", "MinMax"),
  low_dim_method = c("PCA", "t-SNE"),
  perplexity = 30,
  seed = 123
)
```

**Arguments**
- `data`: the `feature_calculations` object containing the raw feature matrix produced by `calculate_features`
- `method`: a rescaling/normalising method to apply. Defaults to "z-score"
- `low_dim_method`: the low dimensional embedding method to use. Defaults to "PCA"
- `perplexity`: the perplexity hyperparameter to use if t-SNE algorithm is selected. Defaults to 30
- `seed`: fixed number for R’s random number generator to ensure reproducibility

**Value**
object of class `low_dimension`
robustsigmoid_scaler

Author(s)
Trent Henderson

Description

\[ z_i = \left[ 1 + \exp \left( \frac{x_i - \text{median}(x)}{\text{IQR}(x)/1.35} \right) \right]^{-1} \]

Usage

robustsigmoid_scaler(x, unitInt = TRUE)

Arguments

x numeric vector
unitInt Boolean whether to rescale into unit interval \([0,1]\). Defaults to TRUE

Value

c numeric vector

Author(s)

Trent Henderson

References

**sigmoid_scaler**

*Rescales a numeric vector using a Sigmoidal transformation*

**Description**

\[ z_i = \left[ 1 + \exp\left( -\frac{x_i - \mu}{\sigma} \right) \right]^{-1} \]

**Usage**

`sigmoid_scaler(x, unitInt = TRUE)`

**Arguments**

- `x` numeric vector
- `unitInt` Boolean whether to rescale into unit interval \([0,1]\). Defaults to `TRUE`.

**Value**

numeric vector

**Author(s)**

Trent Henderson

---

**simData**

*Sample of randomly-generated time series to produce function tests and vignettes*

**Description**

The variables include:

**Usage**

`simData`

**Format**

A tidy data frame with 4 variables:

- `id` Unique identifier for the time series
- `timepoint` Time index
- `values` Value
- `process` Group label for the type of time series
Description
Tools for Handling Extraction of Features from Time-series

zscore_scaler
Rescales a numeric vector into z-scores and then into the unit interval [0,1]

Description
\[ z_i = \frac{x_i - \mu}{\sigma} \]

Usage
zscore_scaler(x, unitInt = TRUE)

Arguments
- x: numeric vector
- unitInt: Boolean whether to rescale into unit interval [0,1]. Defaults to TRUE

Value
numeric vector

Author(s)
Trent Henderson
Index

* datasets
  feature_list, 6
  simData, 16

calculate_features, 2
check_vector_quality, 4
compute_top_features, 4

feature_list, 6
fit_multi_feature_classifier, 7
fit_single_feature_classifier, 9

init_theft, 10

minmax_scaler, 11

normalise, 11

plot.feature_calculations, 12
plot.low_dimension, 13
process_hctsa_file, 13

reduce_dims, 14
robustsigmoid_scaler, 15

sigmoid_scaler, 16
simData, 16

theft, 17

zscore_scaler, 17