Package ‘oddstream’

October 14, 2022

Type Package
Title Outlier Detection in Data Streams
Version 0.5.0
Depends R (>= 3.4.0)
Maintainer Priyanga Dilini Talagala <pritalagala@gmail.com>
Description We proposes a framework that provides real time support for early detection of
anomalous series within a large collection of streaming time series data. By definition, anomalies
are rare in comparison to a system's typical behaviour. We define an anomaly as an observation that
is very unlikely given the forecast distribution. The algorithm first forecasts a boundary for the
system's typical behaviour using a representative sample of the typical behaviour of the system. An
approach based on extreme value theory is used for this boundary prediction pro-
cess. Then a sliding
window is used to test for anomalous series within the newly arrived collection of series. Feature
based representation of time series is used as the input to the model. To cope with concept drift,
the forecast boundary for the system's typical behaviour is updated periodically. More details
regarding the algorithm can be found in Talagala, P. D., Hyndman, R. J., Smith-Miles, K., et al.

BugReports https://github.com/pridiltal/oddstream/issues
License GPL-3
LazyData true
RoxygenNote 6.1.1
Imports pcaPP, stats, ggplot2, ks, MASS, RcppRoll, mgcv, moments, 
RColorBrewer, mvtsplot, tibble, reshape, dplyr, graphics, 
tidyR, kernlab, magrittr
Encoding UTF-8
Suggests testthat, tidyverse
NeedsCompilation no
Author Priyanga Dilini Talagala [aut, cre],
Rob J. Hyndman [ths],
Kate Smith-Miles [ths]
Repository CRAN
Date/Publication 2019-12-16 22:00:03 UTC
R topics documented:

anomalous_stream ................................................. 2
extract_tsfeatures ................................................. 2
find_odd_streams .................................................. 4
get_pc_space ....................................................... 5
gg_featurespace ................................................... 6
oddstream .......................................................... 7
set_outlier_threshold ............................................... 8

Index

anomalous_stream .................................................. 10

anomalous_stream

Multivariate timeseries dataset with an anomalous event.

Description
A multivariate time series dataset with some anomalous series. These time series are with noisy signals.

Usage
anomalous_stream

Format
A data frame with 640 series each with 1459 time points.

extract_tsfeatures

Extract features from a collection of time series

Description
This function extract time series features from a collection of time series. This is a modification of tsmeasures function of anomalous package package.

Usage
extract_tsfeatures(y, normalise = TRUE, width = ifelse(frequency(y) > 1, frequency(y), 10), window = width)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>A multivariate time serie</td>
</tr>
<tr>
<td>normalise</td>
<td>If TRUE, each time series is scaled to be normally distributed with mean 0 and sd 1</td>
</tr>
<tr>
<td>width</td>
<td>A window size for variance change, level shift and lumpiness</td>
</tr>
<tr>
<td>window</td>
<td>A window size for KLscore</td>
</tr>
</tbody>
</table>
Value

An object of class features with the following components:

- **mean**: Mean
- **variance**: Variance
- **lumpiness**: Variance of annual variances of remainder
- **lshift**: Level shift using rolling window
- **vchange**: Variance change
- **linearity**: Strength of linearity
- **curvature**: Strength of curvature
- **spikiness**: Strength of spikiness
- **season**: Strength of seasonality
- **peak**: Strength of peaks
- **trough**: Strength of trough
- **BurstinessFF**: Burstiness of time series using Fano Factor
- **minimum**: Minimum value
- **maximum**: Maximum value
- **rmeaniqmean**: Ratio between interquartile mean and the arithmetic mean
- **moment3**: Third moment
- **highlowmu**: Ratio between the means of data that is below and upper the global mean

References


See Also

- `find_odd_streams`, `get_pc_space`, `set_outlier_threshold`, `gg_featurespace`

Examples

```r
mvtsplot::mvtsplot(anomalous_stream, levels=8, gcol=2, norm="global")
features <- extract_tsfeatures(anomalous_stream[500:550, ])
plot.ts(features[, 1:10])
```
find_odd_streams

Detect outlying series within a collection of streaming time series

Description

This function detects outlying series within a collection of streaming time series. A sliding window is used to handle streaming data. In the presence of concept drift, the forecast boundary for the system's typical behavior can be updated periodically.

Usage

\[
\text{find_odd_streams}(\text{train_data}, \text{test_stream}, \text{update_threshold} = \text{TRUE}, \\
\text{window_length} = \text{nrow(train_data)}, \text{window_skip} = \text{window_length}, \\
\text{concept_drift} = \text{FALSE}, \text{trials} = 500, \text{p_rate} = 0.001, \\
\text{cd_alpha} = 0.05)
\]

Arguments

- **train_data**: A multivariate time series data set that represents the typical behavior of the system.
- **test_stream**: A multivariate streaming time series data set to be tested for outliers.
- **update_threshold**: If TRUE, the threshold value to determine outlying series is updated. The default value is set to TRUE.
- **window_length**: Sliding window size (Ideally this window length should be equal to the length of the training multivariate time series data set that is used to define the outlying threshold).
- **window_skip**: The number of steps the window should slide forward. The default is set to window_length.
- **concept_drift**: If TRUE, the outlying threshold will be updated after each window. The default is set to FALSE.
- **trials**: Input for set_outlier_threshold function. Default value is set to 500.
- **p_rate**: False positive rate. Default value is set to 0.001.
- **cd_alpha**: Significance level for the test of non-stationarity.

Value

A list with components

- **out_matrix**: The indices of the outlying series in each window.
- **p_value**: p-value for the two sample comparison test for concept drift detection.
- **anom_threshold**: Anomalous threshold.

For each window a plot is also produced on the current graphic device.
References


See Also

`extract_tsfeatures`, `get_pc_space`, `set_outlier_threshold`, `gg_featurespace`

Examples

```r
# Generate training dataset
set.seed(890)
nobs = 250
nts = 100
train_data <- ts(apply(matrix(ncol = nts, nrow = nobs), 2, function(nobs){10 + rnorm(nobs, 0, 3)}))
# Generate test stream with some outliyng series
nobs = 15000
test_stream <- ts(apply(matrix(ncol = nts, nrow = nobs), 2, function(nobs){10 + rnorm(nobs, 0, 3)}))
find_odd_streams(train_data, test_stream, trials = 100)

# Considers the first window of the data set as the training set and the remaining as
# the test stream
train_1data <- anomalous_stream[1:100,]
test_stream <- anomalous_stream[101:1456,]
find_odd_streams(train_data, test_stream, trials = 100)
```

def define_feature_space(features, robust = TRUE, kpc = 2)

Description

Define a two dimensional feature space using the first two principal components generated from the features matrix returned by `extract_tsfeatures`

Usage

`get_pc_space(features, robust = TRUE, kpc = 2)`
Arguments

features Feature matrix returned by `extract_tsfeatures`
robust If TRUE, a robust PCA will be used on the feature matrix.
kpc Desired number of components to return.

Value

It returns a list with class ’pcattributes’ containing the following components:

pcnorm The scores of the first kpc principal components
center, scale The centering and scaling used
rotation the matrix of variable loadings (i.e., a matrix whose columns contain the eigen-vectors). The function `princomp` returns this in the element loadings.

See Also

`PCAproj`, `prcomp`, `find_odd_streams`, `extract_tsfeatures`, `set_outlier_threshold`, `gg_featurespace`

Examples

```r
features <- extract_tsfeatures(anomalous_stream[1:100, 1:100])
pc <- get_pc_space(features)
```

---

`gg_featurespace` Produces a ggplot object of two dimensional feature space.

Description

Create a ggplot object of two dimensional feature space using the first two principal components returned by `get_pc_space`.

Usage

`gg_featurespace(object, ...)`

Arguments

object Object of class “pcoddstream”.

... Other plotting parameters to affect the plot.

Value

A ggplot object of two dimensional feature space.
oddstream: A package for Outlier Detection in Data Streams

Description

Rapid advances in hardware technology have enabled a wide range of physical objects, living beings and environments to be monitored using sensors attached to them. Over time these sensors generate streams of time series data. Finding anomalous events in streaming time series data has become an interesting research topic due to its wide range of possible applications such as: intrusion detection, water contamination monitoring, machine health monitoring, etc. This package proposes a framework that provides real time support for early detection of anomalous series within a large collection of streaming time series data. By definition, anomalies are rare in comparison to a system’s typical behaviour. We define an anomaly as an observation that is very unlikely given the forecast distribution. The proposed framework first forecasts a boundary for the system’s typical behaviour using a representative sample of the typical behaviour of the system. An approach based on extreme value theory is used for this boundary prediction process. Then a sliding window is used to test for anomalous series within the newly arrived collection of series. Feature based representation of time series is used as the input to the model. To cope with concept drift, the forecast boundary for the system’s typical behaviour is updated periodically. More details regarding the algorithm can be found in Talagala, P. D., Hyndman, R. J., Smith-Miles, K., et al. (2019) DOI:10.1080/10618600.2019.1617160.

Note

The name oddstream comes from Outlier Detection in Data STREAMs

References


See Also

The core functions in this package: find_odd_streams, extract_tsfeatures, get_pc_space, set_outlier_threshold, gg_featurespace

Examples

features <- extract_tsfeatures(anomalous_stream[1:100, 1:100])
pc <- get_pc_space(features)
p <- gg_featurespace(pc)
p + ggplot2::geom_density_2d()
set_outlier_threshold  

**Set a threshold for outlier detection**

**Description**

This function forecasts a boundary for the typical behaviour using a representative sample of the typical behaviour of a given system. An approach based on extreme value theory is used for this boundary prediction process.

**Usage**

```r
set_outlier_threshold(pc_pcnorm, p_rate = 0.001, trials = 500)
```

**Arguments**

- `pc_pcnorm`: The scores of the first two principal components returned by `get_pc_space`
- `p_rate`: False positive rate. Default value is set to 0.001
- `trials`: Number of trials to generate the extreme value distribution. Default value is set to 500.

**Value**

Returns a threshold to determine outlying series in the next window consists with a collection of time series.

**References**


**See Also**

- `find_odd_streams`, `extract_tsfeatures`, `get_pc_space`, `gg_featurespace`

**Examples**

```r
# Generate training dataset
set.seed(123)
nobs <- 500
nts <- 50
train_data <- ts(apply(matrix(ncol = nts, nrow = nobs), 2, function(nobs){10 + rnorm(nobs, 0, 3)}))
features <- extract_tsfeatures(train_data)
pc <- get_pc_space(features)
threshold <- set_outlier_threshold(pc$pcnorm)
```
set_outlier_threshold

threshold$threshold_fnx
Index

* datasets
  anomalous_stream, 2
anomalous_stream, 2
extract_tsfeatures, 2, 5–8
find_odd_streams, 3, 4, 6–8
get_pc_space, 3, 5, 6–8
gg_featurespace, 3, 5, 6, 7, 8
oddstream, 7
oddstream-package (oddstream), 7
PCAproj, 6
prcomp, 6
set_outlier_threshold, 3, 5–7, 8