Package ‘OSTSC’

December 4, 2017

Title  Over Sampling for Time Series Classification

Version  0.0.1

Author  Matthew Dixon [ctb],
        Diego Klabjan [ctb],
        Lan Wei [aut, trl, cre]

Maintainer  Lan Wei <lweicdsor@gmail.com>

Description  Oversampling of imbalanced univariate time series classification data
using integrated ESPO and ADASYN methods. Enhanced Structure Preserving Oversampling
(ESPO) is used to generate a large percentage of the synthetic minority samples
from univariate labeled time series under the modeling assumption that the predictors
are Gaussian. ESPO estimates the covariance structure of the minority-class samples
and applies a spectral filter to reduce noise. Adaptive Synthetic (ADASYN) sampling
approach is a nearest neighbor interpolation approach which is subsequently applied
to the ESPO samples. This code is ported from a ‘MATLAB’ implementation by Cao et al.
<doi:10.1109/TKDE.2013.37> and adapted for use with Recurrent Neural Networks
implemented in 'TensorFlow'.

Depends  R (>= 3.2.3)

License  GPL-3

URL  https://github.com/lweicdsor/OSTSC

Encoding  UTF-8

LazyData  true

RoxygenNote  6.0.1.9000

Imports  fields, MASS, stats, utils, parallel, doParallel, doSNOW, foreach

Suggests  knitr, rmarkdown, keras, dummies, rlist, pROC, devtools,
          knitr, xtics, testthat, xts

VignetteBuilder  knitr

NeedsCompilation  no

Repository  CRAN

Date/Publication  2017-12-04 15:20:31 UTC
R topics documented:

- Dataset_Adiac
- Dataset_ECG
- Dataset_ElectricalDevices
- Dataset_HFT
- Dataset_HFT300
- Dataset_MHEALTH
- Dataset_Synthetic_Control
- OSTSC

Index

<table>
<thead>
<tr>
<th>Dataset_Adiac</th>
<th>The automatic diatoms identification.</th>
</tr>
</thead>
</table>

Description

The data is collected from a pilot study on automatic identification of diatoms (unicellular algae) from images.

Usage

`data(Dataset_Adiac)`

Format

A dataset with 781 observations and a sequence length of 176, with a single sequence per row.

- The y data is the class label (1 or 0).
- The x data constructs time series sequences (numeric).
- The training dataset contains 390 observations.
- The testing dataset contains 391 observations.

Details

The dataset originally had 37 classes. This built-in data sets one class as the positive class (class 1) and all others are set to the negative class (class 0) to form a highly imbalanced dataset.

Source

Dataset_ECG

The Electrocardiogram dataset.

Description
This dataset recorded heartbeats from the patient with severe congestive heart failure.

Usage
Dataset_ECG()

Format
A dataset with 5000 observations of sequence length 140, with a single sequence per row.
The y data is labeled as 1,3,4,5.
The x data constructs time series sequences (numeric).

Details
The dataset was pre-processed on extracting heartbeats sequences and setting class values from automated annotation.

Value
ecg: the dataset ECG

References


Dataset_ElectricalDevices

The Electrical Devices dataset.

Description
This dataset is taken from research Powering the Nation.
Dataset_HFT

Usage

Dataset_ElectricalDevices()

Format

A dataset with 5527 observations of sequence length 96, with a single sequence per row.
The y data is labeled as 5,6.
The x data constructs time series sequences (numeric).

Details

Powering the Nation collected behavioural data about how consumers use electricity within the
home to help reduce the UK’s carbon footprint.

Value

eds: the dataset ElectricalDevices

References

Electricity Usage Profiles. In: Yin H., Wang W., Rayward-Smith V. (eds) Intelligent Data Engineer-
ing and Automated Learning - IDEAL 2011. IDEAL 2011. Lecture Notes in Computer Science,

Dataset_HFT

The high frequency trading data.

Description

This dataset is a random subset of a high frequency trading dataset used to assess the performace of
RNNs for prediction (Dixon, 2017).

Usage

Dataset_HFT()

Format

A dataset with 30000 observations of sequence length = 10, with a single sequence per row.
The y data is labeled as -1,0,1.
The x data constructs time series sequences (numeric).
Details

The feature represents the instantaneous liquidity imbalance using the best bid to ask ratio. The labels represent the next-event mid-price movement - Y=1 is an up-tick, Y=-1 is a down-tick and Y=0 represents no-movement. The time series sequences length is set to 10. In this package, the class 1 and -1 observations are random selected to yield 1200 non-zero observations, while class 0 has 28800 observations. Observations are ordered chronologically.

Value

hft: the dataset HFT

References

Description

The MHEALTH (Mobile Health) dataset benchmarks techniques for human behavior analysis based on multimodal body sensing.

Usage

Dataset_MHEALTH()

Format

A time series data with multi-classes and multi-features.

#Activities: 12
#Sensor devices: 3
#Subjects: 10

The activity set is listed in the following:
L1: Standing still (1 min)
L2: Sitting and relaxing (1 min)
L3: Lying down (1 min)
L4: Walking (1 min)
L5: Climbing stairs (1 min)
L6: Waist bends forward (20x)
L7: Frontal elevation of arms (20x)
L8: Knees bending (crouching) (20x)
L9: Cycling (1 min)
L10: Jogging (1 min)
L11: Running (1 min)
L12: Jump front & back (20x)

The meaning of each column is detailed next:
Column 1: acceleration from the chest sensor (X axis)
Column 2: acceleration from the chest sensor (Y axis)
Column 3: acceleration from the chest sensor (Z axis)
Column 4: electrocardiogram signal (lead 1)
Column 5: electrocardiogram signal (lead 2)
Column 6: acceleration from the left-ankle sensor (X axis)
Column 7: acceleration from the left-ankle sensor (Y axis)
Column 8: acceleration from the left-ankle sensor (Z axis)
Column 9: gyro from the left-ankle sensor (X axis)
Column 10: gyro from the left-ankle sensor (Y axis)
Column 11: gyro from the left-ankle sensor (Z axis)
Column 12: magnetometer from the left-ankle sensor (X axis)
Column 13: magnetometer from the left-ankle sensor (Y axis)
Column 14: magnetometer from the left-ankle sensor (Z axis)
Column 15: acceleration from the right-lower-arm sensor (X axis)
Column 16: acceleration from the right-lower-arm sensor (Y axis)
Column 17: acceleration from the right-lower-arm sensor (Z axis)
Column 18: gyro from the right-lower-arm sensor (X axis)
Column 19: gyro from the right-lower-arm sensor (Y axis)
Column 20: gyro from the right-lower-arm sensor (Z axis)
Column 21: magnetometer from the right-lower-arm sensor (X axis)
Column 22: magnetometer from the right-lower-arm sensor (Y axis)
Column 23: magnetometer from the right-lower-arm sensor (Z axis)
Column 24: Label (0 for the null class)

In this dataset, for a simple example displaying, only subject 1-5 and feature 12 (magnetometer from the left-ankle sensor (X axis)) are used, and the dataset is reformatted to binary class. Class 11 is set as positive, others as negative. The time series sequences length uses 30. Each sequence occurs in one line.

Details

Recordings of body motion for ten volunteers performing several physical activities. Sensors are placed on the subject’s chest, right wrist and left ankle are used to measure the motion experienced by diverse body parts, namely, acceleration, rate of turn and magnetic field orientation. The sensor positioned on the chest also provides 2-lead ECG measurements, which can be potentially used for basic heart monitoring, checking for various arrhythmias or looking at the effects of exercise on the ECG.

Value

mhealth: the dataset MHEALTH

References


Dataset_Synthetic_Control

*The synthetically generated control charts.*

**Description**

The data represents control charts synthetically generated by the process described in (Alcock and Manolopoulos, 1999).

**Usage**

```r
data(Dataset_Synthetic_Control)
```

**Format**

A dataset with 600 observations and a sequence length of 60, with a single sequence per line.
- The y data is the response (1 or 0).
- The x data constructs time series sequences (numeric).
- Both training and testing datasets contain 300 sequence observations.

**Details**

Class 1 represents 'normal' status, while class 0 represents either 'Cyclic', 'Increasing trend', 'Decreasing trend', 'Upward shift' or Downward shift.

**Source**


---

OSTSC

*Over Sampling for Time Series Classification*

**Description**

Oversample a univariate, multi-modal time series sequence of imbalanced classified data.

**Usage**

```r
OSTSC(sample, label, class, ratio = 1, per = 0.8, r = 1, k = 5,
m = 15, parallel = TRUE, progBar = TRUE)
```
Arguments

- **sample**: Univariate sequence data samples
- **label**: Labels corresponding to samples
- **class**: The number of the classes to be oversampled, starting from the class with the fewest observations, with the default setting to progress to as many classes as possible.
- **ratio**: The oversampling ratio number (>=1) (default = 1)
- **per**: Ratio of weighting between ESPO and ADASYN (default = 0.8)
- **r**: A scalar ratio specifying which level (towards the boundary) we shall push the synthetic data in ESPO (default = 1)
- **k**: Number of nearest neighbours in k-NN (for ADASYN) algorithm (default = 5)
- **m**: Seeds from the positive class in m-NN (for ADASYN) algorithm (default = 15)
- **parallel**: Whether to execute in parallel mode (default = TRUE). (Recommended for datasets with over 30,000 records.)
- **progBar**: Whether to include progress bars (default = TRUE). For ESPO approach, the bar character is |——–|100%. For ADASYN approach, the bar character is |========|100%.

Details

This function balances univariate imbalance time series data based on structure preserving oversampling.

Value

- **sample**: the time series sequences data oversampled
- **label**: the label corresponding to each row of records

References

H. Cao, X.-L. Li, Y.-K. Woon and S.-K. Ng, "Integrated Oversampling for Imbalanced Time Series Classification" IEEE Trans. on Knowledge and Data Engineering (TKDE), vol. 25(12), pp. 2809-2822, 2013


Examples

# This is a simple example to show the usage of OSTSC. See the vignetter for a tutorial
# demonstrating more complex examples.
# loading data
data(Dataset_Synthetic_Control)
# get split feature and label data
train.label <- Dataset_Synthetic_Control$train.y
train.sample <- Dataset_Synthetic_Control$train.x
# the first dimension of the feature set and labels must be the same
# the second dimension of the feature set is the sequence length
dim(train.sample)
dim(train.label)
# check the imbalance ratio of the data
table(train.label)
# oversample class 1 to the same number of observations as class 0
MyData <- OSTSC(train.sample, train.label, parallel = FALSE)
# store the feature data after oversampling
x <- MyData$sample
# store the label data after oversampling
y <- MyData$label
# check the imbalance of the data
table(y)
Index

*Topic datasets
  Dataset_Adiac, 2
  Dataset_HFT300, 5
  Dataset_Synthetic_Control, 8

Dataset_Adiac, 2
Dataset_ECG, 3
Dataset_ElectricalDevices, 3
Dataset_HFT, 4
Dataset_HFT300, 5
Dataset_MHEALTH, 6
Dataset_Synthetic_Control, 8

OSTSC, 8