Package 'fmf'

September 3, 2020

Type Package

Title Fast Class Noise Detector with Multi-Factor-Based Learning

Version 1.1.1

Date 2020-08-07

Author Wanwan Zheng [aut, cre],
Mingzhe Jin [aut],
Lampros Mouselimis [ctb, cph]

Maintainer Wanwan Zheng <teiwanwan@gmail.com>

Description
A fast class noise detector which provides noise score for each observations. The package takes advantage of 'RcppArmadillo' to speed up the calculation of distances between observations.

License MIT + file LICENSE

Encoding UTF-8

SystemRequirements libarmadillo: apt-get install -y libarmadillo-dev (deb), libblas: apt-get install -y libblas-dev (deb), liblapack: apt-get install -y liblapack-dev (deb), libarpack++2: apt-get install -y libarpack++2-dev (deb), gfortran: apt-get install -y gfortran (deb)

LazyData TRUE

Depends R(>= 2.10.0)

Imports Rcpp, caret, solitude, kernlab, C50, e1071, FactoMineR, dplyr, factoextra, ggplot2

LinkingTo Rcpp, RcppArmadillo

Suggests testthat, covr, knitr, rmarkdown

RoxygenNote 7.1.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-09-03 07:32:12 UTC

R topics documented:

`australian` *Australian Credit Approval*

Description

This is the famous Australian Credit Approval dataset, originating from the StatLog project. It concerns credit card applications. All attribute names and values have been changed to meaningless symbols to protect the confidentiality of the data.

Usage

```
data(australian)
```

Format

A data frame with 690 Instances and 15 attributes (including the class attribute, "Class")

Details

There are 6 numerical and 8 categorical attributes, all normalized to [-1,1]. The original formatting was as follows: A1: A,B class attribute (formerly: +,-) A2: 0,1 CATEGORICAL (formerly: a,b) A3: continuous. A4: continuous. A5: 1,2,3 CATEGORICAL (formerly: p,g,gg) A6: 1, 2,3,4,5, 6,7,8,9,10,11,12,13,14 CATEGORICAL (formerly: ff,d,i,k,j,aa,m,c,w, e, q, r,cc, x) A7: 1, 2,3, 4,5,6,7,8,9 CATEGORICAL (formerly: ff,dd,j,bb,v,n,o,h,z) A8: continuous. A9: 1, 0 CATEGORICAL (formerly: t, f) A10: 1, 0 CATEGORICAL (formerly: t, f) A11: continuous. A12: 1, 0 CATEGORICAL (formerly t, f) A13: 1, 2, 3 CATEGORICAL (formerly: s, g, p) A14: continuous. A15: continuous.

Source

Confidential. Donated by Ross Quinlan

References

[LibSVM] (https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html), UCI - 1987

Examples

```
data(australian)

X = australian[, -1]

y = australian[, 1]
```

`fmf` *Fast Class Noise Detector with Multi-Factor-Based Learning*

Description

This function computes the noise score for each observation

Usage

```
fmf(x, ...)

## S3 method for class 'formula'
fmf(formula, data, ...)

## Default S3 method:
fmf(
  x,
  knn = 5,
  classColumn = 1,
  boxplot_range = 1,
  iForest = TRUE,
  threads = 1,
  ...
)
```

Arguments

`...`	optional parameters to be passed to other methods.
`formula`	a formula describing the classification variable and the attributes to be used.
`data, x`	data frame containing the tranining dataset to be filtered.
`knn`	total number of nearest neighbors to be used.The default is 5.
`classColumn`	positive integer indicating the column which contains the (factor of) classes. By default, a dataframe built from 'data' using the variables indicated in 'formula' and The first column is the response variable, thus no need to define the classColumn.
`boxplot_range`	range of box and whisker diagram. The dafault is 1.
`iForest`	compute iForest score or not. The dafault is TRUE.
`threads`	the number of cores to be used in parallel.

Value

an object of class `filter`, which is a list with four components:

- `cleanData` is a data frame containing the filtered dataset.
- `remIdx` is a vector of integers indicating the indexes for removed instances (i.e. their row number with respect to the original data frame).
- `noise_score` is a vector of values indicating the optential of being a noise.
- `call` contains the original call to the filter.

Author(s)

Wanwan Zheng

Examples

```
data(iris)
out = fmf(Species~.,iris)
```

`iris` *Iris Data Set*

Description

This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. (See Duda & Hart, for example.) The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

Usage

```
data(iris)
```

Format

A data frame with 150 Instances and 4 attributes (including the class attribute, "Species") In this package, the iris dataset has been normalized by the max-min normalization.

Details

Fisher,R.A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to Mathematical Statistics" (John Wiley, NY, 1950).

Predicted attribute: class of iris plant.

This is an exceedingly simple domain.

This data differs from the data presented in Fishers article (identified by Steve Chadwick, spchadwick '@' espeedaz.net). The 35th sample should be: 4.9,3.1,1.5,0.2,"setosa" where the error is in the fourth feature. The 38th sample: 4.9,3.6,1.4,0.1,"setosa" where the errors are in the second and third features.

Source

Creator:

R.A. Fisher

Donor:

Michael Marshall

References

https://archive.ics.uci.edu/ml/datasets/iris

Examples

```
data(iris)

X = iris[, -1]

y = iris[, 1]
```

`normalization` *The Max-Min Normalization*

Description

This function normalizes the data using the max-min normalization

Usage

```
normalization(x, margin = 2)
```

Arguments

`x`	the dataset.
`margin`	data is normalized by row (margin = 1) or by column (margin = 2). The default is 2.

Author(s)

Wanwan Zheng

Examples

```
data(ozone)
scaled.data = normalization(ozone[,-1])
ozone.scale = data.frame(y = as.character(ozone[,1]), scaled.data[,-1])
```

---

ozone *Ozone Level Detection Data Set*

Description

Forecasting skewed biased stochastic ozone days: analyses, solutions and beyond, Knowledge and Information Systems, Vol. 14, No. 3, 2008. Discusses details about the dataset, its use as well as various experiments (both cross-validation and streaming) using many state-of-the-art methods.

A shorter version of the paper (does not contain some detailed experiments as the journal paper above) is in: Forecasting Skewed Biased Stochastic Ozone Days: Analyses and Solutions. ICDM 2006: 753-764

Usage

```
data(ozone)
```

Format

A data frame with 2536 Instances and 73 attributes (including the class attribute, "Class": ozone day, normal day)

Details

The following are specifications for several most important attributes that are highly valued by Texas Commission on Environmental Quality (TCEQ). More details can be found in the two relevant papers.

– O 3 - Local ozone peak prediction – Upwind - Upwind ozone background level – EmFactor - Precursor emissions related factor – Tmax - Maximum temperature in degrees F – Tb - Base temperature where net ozone production begins (50 F) – SRd - Solar radiation total for the day – WSa - Wind speed near sunrise (using 09-12 UTC forecast mode) – WSp - Wind speed mid-day (using 15-21 UTC forecast mode)

Source

Kun Zhang zhang.kun05 '@' gmail.com Department of Computer Science, Xavier University of Lousiana

Wei Fan wei.fan '@' gmail.com IBM T.J.Watson Research

XiaoJing Yuan xyuan '@' uh.edu Engineering Technology Department, College of Technology, University of Houston

References

https://archive.ics.uci.edu/ml/datasets/Ozone+Level+Detection

Examples

```
data(ozone)

X = ozone[, -1]

y = ozone[, 1]
```

`plot` *PCA Plot of the Noise Score of Each Individual*

Description

This function plots the noise score for each observation

Usage

```
plot(
  score,
  data,
  cl,
  geom.ind = "text",
  labelsize = 3,
  geom_point_size = 3,
  ...
)
```

Arguments

`score`	a vector of values indicating the optential of being a noise.
`data`	matrix or data frame with no label.
`cl`	factor of true classifications of data set.
`geom.ind`	as geom for observations, which can be set to "text", "point" and "none". The default is "text".

labelsize	size of geom_text.
geom_point_size	size of geom_point and geom_none.
...	optional parameters to be passed to other methods.

Value

an plot of PCA with the noise score of each observation

Author(s)

Wanwan Zheng

Examples

```
data(iris)
out = fmf(Species~.,iris)
plot(out$noise_score, iris[,-1], iris[,1])
```

---

tuning *Tuning For Fast Class Noise Detector with Multi-Factor-Based Learning*

Description

This function tunes the hyper-parameters the threshold and the k of k-NN

Usage

```
tuning(x, ...)

## S3 method for class 'formula'
tuning(formula, data, ...)

## Default S3 method:
tuning(
  x,
  knn_k = seq(3, 7, 2),
  classColumn = 1,
  boxplot_range = seq(0.1, 1.1, 0.2),
  repeats = 10,
  method = "svm",
  iForest = TRUE,
  threads = 1,
  ...
)
```

Arguments

...	Optional parameters to be passed to other methods.
formula	a formula describing the classification variable and the attributes to be used.
data, x	data frame containing the tranining dataset to be filtered.
knn_k	range of the total number of nearest neighbors to be used.The default is 3:5.
classColumn	positive integer indicating the column which contains the (factor of) classes. By default, a dataframe built from 'data' using the variables indicated in 'formula' and The first column is the response variable, thus no need to define the classColumn.
boxplot_range	range of box and whisker diagram. The default is seq(0.8,1.2,0.1).
repeats	the number of cross-validation. The default is 10.
method	the classifier to be used to compute the accuracy. The valid methods are svm (default) and c50.
iForest	compute iForest score or not. The dafault is TRUE.
threads	the number of cores to be used in parallel

Value

An object of class `filter`, which is a list with two components:

- `summary` is the a vector of values when different hyper-parameter is set.
- `call` contains the original call to the filter.

Author(s)

Wanwan Zheng

Examples

```
data(iris)
out = tuning(Species~.,iris)
```

Index