Package 'IRon'

October 29, 2022

Type Package

Title Solving Imbalanced Regression Tasks

Version 0.1.3

Description Imbalanced domain learning has almost exclusively focused on solving classification tasks, where the objective is to predict cases labelled with a rare class accurately. Such a well-defined approach for regression tasks lacked due to two main factors. First, standard regression tasks assume that each value is equally important to the user. Second, standard evaluation metrics focus on assessing the performance of the model on the most common cases. This package contains methods to tackle imbalanced domain learning problems in regression tasks, where the objective is to predict extreme (rare) values.
The methods contained in this package are: 1) an automatic and non-parametric method to obtain such relevance functions; 2) visualisation tools; 3) suite of evaluation measures for optimisation/validation processes; 4) the squared-error relevance area measure, an evaluation metric tailored for imbalanced regression tasks.
More information can be found in Ribeiro and Moniz (2020) <doi:10.1007/s10994-020-05900-9>.

URL https://github.com/nunompmoniz/IRon

BugReports https://github.com/nunompmoniz/IRon/issues

License CC0

Encoding UTF-8

LazyData true

LinkingTo Rcpp

Depends R (>= 2.10)

Imports Rcpp, stats, ggpubr, gridExtra, ggplot2, robustbase, scam

Suggests rpart, e1071, earth, randomForest, mgcv, reshape

RoxygenNote 7.2.1

NeedsCompilation yes

Author Nuno Moniz [cre, aut],
Rita P. Ribeiro [aut],
Miguel Margarido [ctb]

Maintainer Nuno Moniz <nmoniz2@nd.edu>

Repository CRAN

Date/Publication 2022-10-28 23:32:51 UTC

R topics documented:

`accel` *Acceleration*

Description

Dataset with acceleration target value w.r.t. 14 nominal and numerical variables

Usage

```
data(accel)
```

Format

A `"data.frame"` structure with 1732 observations, 3 nominal and 11 numerical predictor variables

Source

UCI Archive

References

Hadi Fanaee-T. and João Gama. Event labeling combining ensemble detectors and background knowledge. Prog. in Art. Int., pages 1-15, 2013. ISSN 2192-6352. (Springer)

Examples

```
data(accel)
head(accel)
```

`eval.stats` *Predictive Modelling Evaluation Statistics*

Description

Evaluation statistics including standard and non-standard evaluation metrics. Returns a structure of data containing the results of several evaluation metrics (both standard and some focused on the imbalanced regression problem).

Usage

```
eval.stats(formula, train, test, y_pred, phi.parms = NULL, cf = 1.5)
```

Arguments

`formula`	A model formula
`train`	A data.frame object with the training data
`test`	A data.frame object with the test set
`y_pred`	A vector with the predictions of a given model
`phi.parms`	The relevance function providing the data points where the pairs of values-relevance are known (use ?phi.control() for more information). If this parameter is not defined, this method will create a relevance function based on the data.frame variable in parameter train. Default is NULL
`cf`	The coefficient used to calculate the boxplot whiskers in the event that a relevance function is not provided (parameter phi.parms)

Value

A list with four slots for the results of standard and relevance-based evaluation metrics

`overall`	Results for standard metrics MAE, MSE and RMSE, along with Pearson's Correlation, bias, variance and the Squared Error Relevance Area metric.

Examples

```
library(IRon)

if(requireNamespace("earth")) {

   data(accel)

   form <- acceleration ~ .

   ind <- sample(1:nrow(accel),0.75*nrow(accel))

   train <- accel[ind,]
   test <- accel[-ind,]
```

```
  ph <- phi.control(accel$acceleration)

  m <- earth::earth(form, train)
  preds <- as.vector(predict(m,test))

  eval.stats(form, train, test, preds)
  eval.stats(form, train, test, preds, ph)
  eval.stats(form, train, test, preds, ph, cf=3) # Focusing on extreme outliers

}
```

NO2Emissions *NO2Emissions*

Description

The data are a subsample of 500 observations from a data set that originate in a study where air pollution at a road is related to traffic volume and meteorological variables, collected by the Norwegian Public Roads Administration. The response variable (column 1) consist of hourly values of the logarithm of the concentration of NO2 (particles), measured at Alnabru in Oslo, Norway, between October 2001 and August 2003. The predictor variables (columns 2 to 8) are the logarithm of the number of cars per hour, temperature 2 meter above ground (degree C), wind speed (meters/second), the temperature difference between 25 and 2 meters above ground (degree C), wind direction (degrees between 0 and 360), hour of day and day number from October 1. 2001.

Usage

```
data(NO2Emissions)
```

Format

A "data.frame" structure with 500 observations, 8 numerical variables

Source

StatLib

Examples

```
data(NO2Emissions)
head(NO2Emissions)
```

phi *Obtain the relevance of data points*

Description

The phi function retrieves the relevance value of the values in a target variable. It does so by resorting to the Piecewise Cubic Hermite Interpolation Polynomial method for interpolating over a set of maximum and minimum relevance points. The notion of relevance is associated with rarity.Nonetheless, this notion may depend on the domain experts knowledge

Usage

```
phi(y, phi.parms = NULL)
```

Arguments

y	The target variable of a given data set
phi.parms	The relevance function providing the data points where the pairs of values-relevance are known

Value

A vector with the relevance values of a given target variable

Examples

```
library(IRon)
data(accel)

ind <- sample(1:nrow(accel),0.75*nrow(accel))

train <- accel[ind,]
test <- accel[-ind,]

ph <- phi.control(train$acceleration)
phis <- phi(test$acceleration,phi.parms=ph)

plot(test$acceleration,phis,xlab="Y",ylab="Relevance")
```

`phi.control` *Generation of relevance function*

Description

This procedure enables the generation of a relevance function that performs a mapping between the values in a given target variable and a relevance value that is bounded by 0 (minimum relevance) and 1 (maximum relevance). This may be obtained automatically (based on the distribution of the target variable) or by the user defining the relevance values of a given set of target values - the remaining values will be interpolated.

Usage

```
phi.control(
  y,
  phi.parms,
  method = phiMethods,
  extr.type = NULL,
  control.pts = NULL,
  asym = TRUE,
  ...
)
```

Arguments

`y`	The target variable of a given data set
`phi.parms`	The relevance function providing the data points where the pairs of values-relevance are known
`method`	The method used to generate the relevance function (extremes or range)
`extr.type`	Type of extremes to be considered: low, high or both (default)
`control.pts`	Parameter required when using 'range' method, representing a 3-column matrix of y-value, corresponding relevance value (between 0 and 1), and the derivative of such relevance value
`asym`	Boolean for assymetric interpolation. Default TRUE, uses adjusted boxplot. When FALSE, uses standard boxplot.
`...`	Misc data to be added to the relevance function

Value

A list with three slots with information concerning the relevance function

`method`	The method used to generate the relevance function (extremes or range)
`npts`	?
`control.pts`	Three sets of values identifying the target value-relevance-derivate for the first low extreme value, the median, and first high extreme value

Examples

```
library(IRon)

data(accel)

ind <- sample(1:nrow(accel),0.75*nrow(accel))

train <- accel[ind,]
test <- accel[-ind,]

ph <- phi.control(train$acceleration); phiPlot(test$acceleration, ph)
ph <- phi.control(train$acceleration, extr.type="high"); phiPlot(test$acceleration, ph)
ph <- phi.control(train$acceleration, method="range",
 control.pts=matrix(c(10,0,0,15,1,0),byrow=TRUE,ncol=3)); phiPlot(test$acceleration, ph)
```

`phiPlot` *Plot of phi versus y and boxplot of y*

Description

The phiPlot function uses a dataset ds containing many y values to produce a line plot of phi versus y and a boxplot of y, and aligns them, one above the other. The first extreme value on either side of the boxplot should correspond to the point where phi becomes exactly 1 on the line plot. This function is dependent on the robustbase, ggplot2 and ggpubr packages, and will not work without them.

Usage

```
phiPlot(ds, phi.parms = NULL, limits = NULL, xlab = "y", ...)
```

Arguments

`ds`	Dataset of y values
`phi.parms`	The relevance function providing the data points where the pairs of values-relevance are known. Default is NULL
`limits`	Vector with values to draw limits. Default is NULL
`xlab`	Label of the x axis. Default is y
`...`	Extra parameters when deriving the relevance function

Value

A line plot of phi versus y, as well as a boxplot of y

Examples

```
ds <- rnorm(1000, 30, 10); phi.parms <- phi.control(ds); phiPlot(ds,phi.parms)
ds <- rpois(100,3); phiPlot(ds)
```

ser *Non-Standard Evaluation Metrics*

Description

Obtains the squared error of predictions for a given subset of relevance

Usage

```
ser(trues, preds, phi.trues = NULL, ph = NULL, t = 0)
```

Arguments

trues	Target values from a test set of a given data set. Should be a vector and have the same size as the variable preds
preds	Predicted values given a certain test set of a given data set. Should be a vector and have the same size as the variable preds
phi.trues	Relevance of the values in the parameter trues. Use ??phi() for more information. Defaults to NULL
ph	The relevance function providing the data points where the pairs of values-relevance are known. Default is NULL
t	Relevance cut-off. Default is 0.

Details

Squared Error-Relevance Metric (SER)

Value

Squared error for for cases where the relevance of the true value is greater than t (SERA)

Examples

```
library(IRon)
library(rpart)

if(requireNamespace("rpart")) {

   data(accel)

   form <- acceleration ~ .

   ind <- sample(1:nrow(accel),0.75*nrow(accel))

   train <- accel[ind,]
   test <- accel[-ind,]

   ph <- phi.control(accel$acceleration)
```

```
  m <- rpart::rpart(form, train)
  preds <- as.vector(predict(m,test))

  trues <- test$acceleration
  phi.trues <- phi(test$acceleration,ph)

  ser(trues,preds,phi.trues)

}
```

`sera` *Squared Error-Relevance Area (SERA)*

Description

Computes an approximation of the area under the curve described by squared error of predictions for a sequence of subsets with increasing relevance

Usage

```
sera(
  trues,
  preds,
  phi.trues = NULL,
  ph = NULL,
  pl = FALSE,
  m.name = "Model",
  step = 0.001,
  return.err = FALSE,
  norm = FALSE
)
```

Arguments

`trues`	Target values from a test set of a given data set. Should be a vector and have the same size as the variable preds
`preds`	Predicted values given a certain test set of a given data set. Should be a vector and have the same size as the variable preds
`phi.trues`	Relevance of the values in the parameter trues. Use ??phi() for more information. Defaults to NULL
`ph`	The relevance function providing the data points where the pairs of values-relevance are known. Default is NULL
`pl`	Boolean to indicate if an illustration of the curve should be provided. Default is FALSE
`m.name`	Name of the model to be appended in the plot title

step	Relevance intervals between 0 (min) and 1 (max). Default 0.001
return.err	Boolean to indicate if the errors at each subset of increasing relevance should be returned. Default is FALSE
norm	Normalize the SERA values for internal optimisation only (TRUE/FALSE)

Value

Value for the area under the relevance-squared error curve (SERA)

Examples

```
library(IRon)
library(rpart)

if(requireNamespace("rpart")) {

   #' data(accel)

   form <- acceleration ~ .

   ind <- sample(1:nrow(accel),0.75*nrow(accel))

   train <- accel[ind,]
   test <- accel[-ind,]

   ph <- phi.control(accel$acceleration)

   m <- rpart::rpart(form, train)
   preds <- as.vector(predict(m,test))

   trues <- test$acceleration
   phi.trues <- phi(test$acceleration,ph)

   sera(trues,preds,phi.trues)
   sera(trues,preds,phi.trues,pl=TRUE, m.name="Regression Trees")
   sera(trues,preds,phi.trues,pl=TRUE, return.err=TRUE)

}
```

Index