Package ‘RatingScaleReduction’

October 12, 2022

Type Package

Title Rating Scale Reduction Procedure

Version 1.4

Date 2021-01-21

Author Waldemar W. Koczokodaj, Feng Li, Alicja Wolny-Dominiak

Maintainer Alicja Wolny-Dominiak <alicja.wolny-dominiak@ue.katowice.pl>

Depends pROC, ggplot2

Description Describes a new procedure of reducing items in a rating scale called Rating Scale Reduction (RSR). The new stop criterion in RSR procedure is added (stop global max). The function order is replaced by sort.list.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2021-01-21 09:30:02 UTC

R topics documented:

RatingScaleReduction-package ............................................. 2
CheckAttr4Inclusion ...................................................... 2
diffExamples ............................................................... 4
grayExamples ............................................................... 5
rsr .......................................................................... 6
startAuc ................................................................. 8
totalAuc ................................................................. 9

Index 11
RatingScaleReduction-package

*Rating Scale Reduction Procedure*

**Description**

This package describes a procedure of reducing items in a rating scale. It was published in the reference included in this description. The method was proposed by Waldemar W. Koczkodaj and published by a sizable collaboration coordinated by him.

**Author(s)**

Waldemar W. Koczkodaj, Feng Li, Alicja Wolny-Dominiak
Maintainer: Alicja Wolny-Dominiak

**References**

1. W.W. Koczkodaj, T. Kakiashvili, A. Szymanska, J. Montero-Marin, R. Araya, J. Garcia-Campayo, K. Rutkowski, D. Strzalka, How to reduce the number of rating scale items without predictability loss? Scientometrics, 909(2):581-593 (open access), 2017
   https://link.springer.com/article/10.1007/s11192-017-2283-4


   https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-12-77


---

**CheckAttr4Inclusion**

*Check the next attribute for possible inclusion into AUC*

**Description**

The attribute checked for AUC before it is added to the running total. The running total is used with the class (decision attribute) to compute AUC. The next attribute is added to the sequence of attributes having the MAX total AUC.
CheckAttr4Inclusion

Usage

CheckAttr4Inclusion(attribute, D, plotCheck=FALSE, method=c("delong", "bootstrap", "venkatraman", "sensitivity", "specificity"), boot.n, alternative = c("two.sided", "less", "greater"))

Arguments

attribute    a matrix or data.frame containing attributes
D            the decision vector
plotCheck    If TRUE the plot with two ROC curves is created
method       the method to useas in the function roc.test{pROC}
boot.n       boostrap replication number
alternative  the alternative hypothesis

Value

test        the result of the roc.test as in the function roc.test from the package pROC

Author(s)

Waldemar W. Koczkodaj, Feng Li, Alicja Wolny-Dominiak

References


   https://link.springer.com/article/10.1007/s11192-017-2283-4

Examples

#creating the matrix of attributes and the decision vector
#must be as.numeric()
data(aSAH)
attach(aSAH)
is.numeric(aSAH)

attribute <- data.frame(as.numeric(gender),
                        as.numeric(age), as.numeric(wfns), as.numeric(s100b), as.numeric(ndka))
colnames(attribute) <- c("a1", "a2", "a3", "a4", "a5")
decision <- as.numeric(outcome)

#deLong test, two-side alternative hipotesis
diffExamples

The number of different (unique) examples in a dataset

Description

Datasets often contain replications. In particular, one example may be replicated n times, where n is the total number of examples, so that there are no other examples. Such situation would deviate computations and should be early detected. Ideally, no example should be replicated but if the rate is small, we can progress to computing AUC.

Usage

diffExamples(attribute)

Arguments

attribute  a matrix or data.frame containing attributes

Value

total.examples  a number of examples in a data
diff.examples  a number of different examples in a data
dup.exapmles  a number of duplicate examples in a data

Author(s)

Waldemar W. Koczkodaj, Feng Li, Alicja Wolny-Dominiak

Examples

#creating the matrix of attributes and the decision vector
#must be as.numeric()
data(aSAH)
attach(aSAH)
is.numeric(aSAH)

attribute <- data.frame(as.numeric(gender),
as.numeric(age), as.numeric(wfns), as.numeric(s100b), as.numeric(ndka))
colnames(attribute) <- c("a1", "a2", "a3", "a4", "a5")

#show the number of different examples
diffExamples(attribute)
grayExamples  Examples belonging to both classes

Description
A subset of data with examples having identical values on all attributes (excluding the class attribute also called the decision attribute which is different and has two permitted values: positive and negative)

Usage
grayExamples(attribute, D)

Arguments
attribute  a matrix or data.frame containing attributes
D          the decision vector

Value
1          a list of pairs of identical examples on all attributes

Author(s)
Waldemar W. Koczkodaj, Alicja Wolny-Dominiak

Examples
#generate data
a=c(); attribute=c()
for (i in 1:3){
a <-sample(c(1,2,3), 100, replace=TRUE)
attribute <-cbind(attribute, a)
attribute=data.frame(attribute)
}
colnames(attribute)=c("a1", "a2", "a3")
names(attribute)

decision=sample(c(0,1), 100, replace=TRUE)

#check examples
grayExamples(attribute, decision)
This package implements a rather sophisticated method published in (Koczkodaj et al., 2017) In essence, it is a stepwise method for maximizing the area under the area (AUC) of receiver operating characteristic (ROC). In this description, data mining terminology will be used:

- examples (observations in statistics),
- variables in statistics,
- class or decision attribute (decision variable may be used statistics).

The implemented algorithm (when reduced to its minimum) comes to using a loop for all attributes (with the class excluded) to compute AUC. Subsequently, attributes are sorted in the descending order by AUC. The attribute with the largest AUC is added to a subset of all attributes (evidently, it cannot be empty since it is supposed to be the minimum subset S of all attributes with the maximum AUC). We keep adding the next in line (according to AUC) attribute to the subset S checking AUC. If it decreases, we stop the procedure. The above procedure can be described by the following algorithm.

Algorithm:

1. compute AUC of all attributes excluding class
2. sort attributes by their AUC in the ascending order
3. select the attribute with the largest AUC to subset S
4. select the next attribute A with the largest AUC to subset S
5. if the AUC of the subset S is larger that AUC of the former AUC then go to 3

There are a lot of checking (e.g., if the dataset is not empty or full of replications) involved.

Usage

rsr(attribute, D, plotRSR = FALSE, method=c('Stop1Max', 'StopGlobalMax'))

Arguments

- attribute: a matrix or data.frame containing attributes
- D: the decision vector
- plotRSR: If TRUE the ROC curve is plotted
- method: the Stop reduction criteria: First Max of AUC or Global Max of AUC, default: 'Stop1Max'
Value

rsr.auc  total AUC of attributes
rsr.label  attribute labels
summary  a summary table

Author(s)

Waldemar W. Koczkodaj, Alicja Wolny-Dominiak

References

   https://link.springer.com/article/10.1007/s11192-017-2283-4


   https://bmcbiinformatics.biomedcentral.com/articles/10.1186/1471-2105-12-77

Examples

```r
#creating the matrix of attributes and the decision vector
#must be as.numeric()
data(aSAH)
attach(aSAH)
is.numeric(aSAH)

attribute <- data.frame(as.numeric(gender),
                        as.numeric(age), as.numeric(wfns), as.numeric(s100b), as.numeric(ndka))
colnames(attribute) <- c("a1", "a2", "a3", "a4", "a5")
decision <- as.numeric(outcome)

#rating scale reduction procedure
rsred <- rsr(attribute, decision, plotRSR=TRUE)
rsred
```
Description

Compute AUC of every single attribute

Usage

\texttt{startAuc(attribute, D)}

Arguments

\texttt{attribute} \hspace{1cm} \text{a matrix or data.frame containing attributes}
\texttt{D} \hspace{1cm} \text{the decision vector}

Value

\texttt{auc} \hspace{1cm} \text{AUC of a single attribute}
\texttt{item} \hspace{1cm} \text{attribute labels}
\texttt{summary} \hspace{1cm} \text{a summary table}

Author(s)

Waldemar W. Koczkodaj, Alicja Wolny-Dominiak

References

   https://link.springer.com/article/10.1007/s11192-017-2283-4

   https://bmcbiinformatics.biomedcentral.com/articles/10.1186/1471-2105-12-77

Examples

\#creating the matrix of attributes and the decision vector
\#must be as.numeric()
data(aSAH)
attach(aSAH)
is.numeric(aSAH)

attribute <- data.frame(as.numeric(gender),
totalAuc

as.numeric(age), as.numeric(wfns), as.numeric(s100b), as.numeric(ndka))
colnames(attribute) <-c("a1", "a2", "a3", "a4", "a5")
decision <-as.numeric(outcome)

#compute AUC of all attributes
start <-startAuc(attribute, decision)
start$summary

---

**totalAuc**  
*AUC of the running total of attributes*

### Description

AUC values are computed for all individual attributes. We sort them in an ascending order. We begin with the attribute having the largest AUC and add to it the second, third,... attribute until AUC of the total of them decreases.

### Usage

```r
totalAuc(attribute, D, plotT = FALSE)
```

### Arguments

- `attribute`: a matrix or data.frame containing attributes
- `D`: the decision vector
- `plotT`: If TRUE the plot is created: x - labels of attributes, y - total AUC in ascending order

### Value

- `ordered.attribute`: ordered attribute matrix
- `total.auc`: total AUC
- `item`: ordered attribute labels
- `summary`: a summary table

### Author(s)

Waldemar W. Koczkodaj, Alicja Wolny-Dominiak
References

https://link.springer.com/article/10.1007/s11192-017-2283-4


https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-12-77

Examples

```r
#creating the matrix of attributes and the decision vector  
#must be as.numeric()  
data(aSAH)  
attach(aSAH)  
is.numeric(aSAH)

attribute <-data.frame(as.numeric(gender),  
as.numeric(age), as.numeric(wfns), as.numeric(s100b), as.numeric(ndka))  
colnames(attribute) <-c("a1", "a2", "a3", "a4", "a5")  
decision <-as.numeric(outcome)

#arrange start AUC in an ascending order and compute total AUC according to  
#Rating Scale Reduction procedure

tot <-totalAuc(attribute, decision, plotT=TRUE)  
tot$summary
```
Index

CheckAttr4Inclusion, 2

diffExamples, 4

grayExamples, 5

RatingScaleReduction
  (RatingScaleReduction-package), 2

RatingScaleReduction-package, 2

rsr, 6

startAuc, 8

totalAuc, 9