Package ‘HUM’

January 20, 2014

Type Package

Title compute HUM value and visualize ROC curves

Version 1.0

Date 2013-10-25

Author Natalia Novoselova, Junxi Wang, Jialiang Li, Frank Pessler, Frank Klawonn

Maintainer Natalia Novoselova <novosVU@mailNru>

Description Tools for computing HUM (Hypervolume Under the Manifold) value to estimate features ability to discriminate the class labels, visualizing the ROC curve for two or three class labels.

License GPL (>= 3)

Imports rgl

Depends R (>= 2.10), gtools, Rcpp (>= 0.10.5)

LinkingTo Rcpp

LazyLoad yes

URL http://public.ostfalia.de/~klawonn/HUM.htm

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-01-20 12:05:33
Description

Functions to calculate AUC (area under curve) value for two classes and HUM (hypervolume under manifold) for more class labels in order to estimate the informativity of features to outcome. Tools for visualizing ROC curve in 2D- and 3D-space.

Details

Package: HUM
Type: Package
Version: 1.0
Date: 2013-10-25
License: GPL (>= 3)

The basic unit of the HUM package is the CalculateHUM_seq function. It will calculate the AUC in case of two class labels and HUM for more than two class labels for the input features. Function CalculateHUM_Ex is the extension of main function and provides the possibility to calculate all the combinations of amountl from all the class labels. Function CalculateHUM_ROC calculates the point coordinates in order to plot the 2D- and 3D-ROC curve, accuracy and the optimal threshold for the classifier (feature). The Functions CalcGene and CalcROC are the auxiliar function to perform the calculation. Function CalcROC calculates the point coordinates of a single feature for two-class or three-class problem, the optimal threshold for the 2-D and 3-D ROC curve and the corresponding feature values, the accuracy of the classifier (feature) for the optimal threshold.

Functions

CalculateHUM_seq Calculate a maximal HUM value and the corresponding permutation of class labels
CalculateHUM_Ex Calculate the HUM values with exhaustive search for specified number of class labels
**CalculateHUM_ROC**  Function to construct and plot the 2D- or 3D-ROC curve

**CalcGene**  Compute the HUM value for one feature

**CalcROC**  Compute the point coordinates to plot the 2D- or 3D-ROC curve

**CalculateHUM_Plot**  Plot the 2D-ROC curve

**Calculate3D**  Plot the 3D-ROC curve

**Dataset**

This package comes with one simulated dataset and a real dataset of 92 patients with 11 features with disease.

**Installing and using**

To install this package, make sure you are connected to the internet and issue the following command in the R prompt:

```
install.packages("HUM")
```

To load the package in R:

```
library(HUM)
```

**Author(s)**

Natalia Novoselova, Frank Pessler

Maintainer: Natalia Novoselova <novos65@mail.ru>

**References**


**See Also**

CRAN packages **pROC**, or Bioconductor’s **roc** for ROC curves.

CRAN packages **Rcpp, gtools, rgl** employed in this package.
Examples

data(sim)

# Compute the HUM value with all possible class label permutation
indexF=c(3,4);
indexClass=2;
label=unique(sim[,indexClass])
indexLabel=label[1:3]
out=CalculateHUM_seq(sim,indexF,indexClass,indexLabel)

# Compute the HUM value with exaustive search of all class label combinations
## Not run: data(sim)
indexF=c(3,4);
indexClass=2;
labels=unique(sim[,indexClass])
amountL=4;
out=CalculateHUM_Ex(sim,indexF,indexClass,labels,amountL)

## End(Not run)
## Calculate the coordinates for 2D- or 3D- ROC curve and the optimal threshold point
## Not run: data(sim)
indexF=names(sim[,c(3),drop = FALSE])
indexClass=2
label=unique(sim[,indexClass])
indexLabel=label[1:3]
out=CalculateHUM_seq(sim,indexF,indexClass,indexLabel)
HUM<-out$HUM
seq<-out$seq
out=CalculateHUM_ROC(sim,indexF,indexClass,indexLabel,seq)

## End(Not run)

CalcGene

Description

This is the auxiliary function of the HUM package. It computes a HUM value for individual feature and returns a “List” object, consisting of HUM value and the best permutation of class labels in “seq” vector. This “seq” vector can be passed to the function CalculateHUM_ROC.

Usage

CalcGene(s_data, seqAll, prodValue)

Arguments

s_data a list, which contains the vectors of sorted feature values for individual class labels.
seqAll a numeric matrix of all the permutations of the class labels, where each row corresponds to individual permutation vector.

prodValue a numeric value, which is the product of the sizes of feature vectors, corresponding to analyzed class labels.

Details

This function’s main job is to compute the maximal HUM value between the all possible permutations of class labels for individual feature, selected for analysis. See the “Value” section to this page for more details.

Value

The data must be provided without missing values in order to process. A returned list consists of the following fields:

- **HUM** a list of HUM values for the specified number of analyzed features
- **seq** a list of vectors, each containing the sequence of class labels

Errors

If there exists NA values for features or class labels no HUM value can be calculated and an error is triggered with message “Values are missing”.

References


See Also

*CalculateHUM_Ex, CalculateHUM_ROC*

Examples

data(sim)
# Basic example
indexf=3;
indexClass=2;
indexLabel=c("Normal","OrthArthr")
s_data=NULL;
prodValue=1;
for(i in 1:length(indexLabel))
{
  index=which(sim[,indexClass]==indexLabel[i])
  vrem=sort(sim[index,indexF])
  s_data=c(s_data,list(vrem))
  prodValue=prodValue*length(index)
}
len=length(indexLabel)
seqAll=permutations(len,len,1:len)
out=CalcGene(s_data, seqAll, prodValue)
CalcROC

**Calculate ROC points**

**Description**

This is the auxiliary function of the HUM package. It computes a point coordinates for plotting ROC curve and returns a “List” object, consisting of sensitivity and specificity values for 2D-ROC curve and 3D-points for 3D-ROC curve, the optimal threshold values with the corresponding feature values and the accuracy of the classifier (feature).

**Usage**

CalcROC(s_data, seq, thresholds)

**Arguments**

- **s_data**: a list, which contains the vectors of sorted feature values for individual class labels.
- **seq**: a numeric vector, containing the particular permutation of class labels.
- **thresholds**: a numeric vector, containing the values of thresholds for calculating ROC curve coordinates.

**Details**

This function’s main job is to compute the point coordinates to plot the 2D- or 3D-ROC curve, the optimal threshold values and the accuracy of classifier. See the “Value” section to this page for more details. The optimal threshold for two-class problem is the pair of sensitivity and specificity values for the selected feature. The optimal threshold for three-class problem is the 3D-point with the coordinates presenting the fraction of the correctly classified data objects for each class. The calculation of the optimal threshold is described in section “Threshold”.

**Value**

The data must be provided without missing values in order to process. A returned list consists of the following fields:

- **Sn**: a specificity values for 2D-ROC construction and the first coordinate for 3D-ROC construction
- **Sp**: a sensitivity values for 2D-ROC construction and the second coordinate for 3D-ROC construction
- **S3**: the third coordinate for 3D-ROC construction
- **optSn**: the optimal specificity value for 2D-ROC construction and the first coordinate of the optimal threshold for 3D-ROC construction
- **optSp**: the optimal sensitivity value for 2D-ROC construction and the second coordinate of the optimal threshold for 3D-ROC construction
- **optS3**: the third coordinate of the optimal threshold for 3D-ROC construction
CalcROC

The optimal threshold value is calculated for two-class problem as the pair \((\text{optSn, optSp})\) corresponding to the maximal value of \(\text{Sn} + \text{Sp}\). According to \((\text{optSn, optSp})\) the corresponding feature threshold value “\text{optThre}” is calculated. The optimal threshold value is calculated for three-class problem as the pair \((\text{optSn, optSp, optS3})\) corresponding to the maximal value of \(\text{Sn} + \text{Sp} + \text{S3}\). According to \((\text{optSn, optSp, optS3})\) the corresponding feature threshold values \(\text{optThre1, optThre2}\) are calculated. The accuracy of the classifier is the mean value of \text{dQuote(optSn, optSp)} for two-class problem and the mean value of \((\text{optSn, optSp, optS3})\) for three-class problem.

Errors

If there exists NA values for features or class labels no HUM value can be calculated and an error is triggered with message “Values are missing”.

References


See Also

CalculateHUM_Ex, CalculateHUM_ROC

Examples

data(sim)
indexF=names(sim[,c(3,4),drop = FALSE])
indexClass=2
label=unique(sim[,indexClass])
indexLabel=label[1:3]
out=CalculateHUM_seq(sim,indexF,indexClass,indexLabel)
HUM<-out$HUM
seq<-out$seq

indexL=NULL
for(i in 1:length(indexLabel)){
  indexL=c(indexL,which(label==indexLabel[i]))
}

indexEach=NULL
Calculate3D

plot the 3D-ROC curve

Description

This is the main function of the HUM package. It plots the 3D-ROC curve using the point coordinates, computed by the function `calculatehum_roc`. Optionally visualizes the optimal threshold point, which gives the maximal accuracy of the classifier(feature) (see `calcroc`).

Usage

`Calculate3D(sel, Sn, Sp, S3, optSn, optSp, optS3, thresholds, HUM, name, print.optim=TRUE)`

Arguments

- `sel` a character value, which is the name of the selected feature.
- `Sn` a numeric vector of the x-coordinates of the ROC curve.
- `Sp` a numeric vector of the y-coordinates of the ROC curve.
- `S3` a numeric vector of the z-coordinates of the ROC curve.
- `optSn` the first coordinate of the optimal threshold
- `optSp` the second coordinate of the optimal threshold
- `optS3` the third coordinate of the optimal threshold
Calculation3D

thresholds a numeric vector with threshold values to calculate point coordinates.
HUM a numeric vector of HUM values, calculated using function.
name a character vector of class labels.
print.optim a boolean parameter to plot the optimal threshold point on the graph. The default value is TRUE.

Details

This function’s main job is to plot the 3D-ROC curve according to the given point coordinates.

Value

The function doesn’t return any value.

Errors

If there exists NA values for specificity or sensitivity values, or HUM values the plotting fails and an error is triggered with message “Values are missing”

References


See Also

CalculateHUM_seq, CalculateHUM_ROC

Examples

data(sim)
indexF=names(sim[,c(3),drop = FALSE])
indexClass=2
label=unique(sim[,indexClass])
indexLabel=label[1:3]
out=CalculateHUM_seq(sim,indexF,indexClass,indexLabel)
HUM<-out$HUM
seq<-out$seq
out=CalculateHUM_ROC(sim,indexF,indexClass,indexLabel,seq)
Calculate3D(indexF, out$Sn, out$Sp, out$S3, out$optSn, out$optSp, out$optS3,
out$thresholds, HUM, indexLabel[seq])
CalculateHUM_Ex  Calculate HUM value

Description

This is the main function of the HUM package. It computes a HUM value and returns a “List” object, consisting of HUM value and the best permutation of class labels in “seq” vector. This “seq” vector can be passed to the function CalculateHUM_ROC.

Usage

CalculateHUM_Ex(data,indexF,indexClass,allLabel,amountL)

Arguments

data  a dataset, a matrix of feature values for several cases, the additional column with class labels is provided. Class labels could be numerical or character values. The maximal number of classes is ten. The indexClass determines the column with class labels.

indexF  a numeric or character vector, containing the column numbers or column names of the analyzed features.

indexClass  a numeric or character value, containing the column number or column name of the class labels.

allLabel  a character vector, containing the column names of the class labels, selected for the analysis.

amountL  a character vector, containing the column names of the class labels, selected for the analysis.

Details

This function’s main job is to compute the maximal HUM value between the all possible permutations of class labels, selected for analysis. See the “Value” section to this page for more details. Before returning, it will call the CalcGene function to calculate the HUM value for each case (object).

Data can be provided in matrix form, where the rows correspond to cases with feature values and class label. The columns contain the values of individual features and the separate column contains class labels. The maximal number of class labels equals 10. The computational efficiency of the function decrease in the case of more than 1000 cases with more than 6 class labels.

Value

The data must be provided without missing values in order to process. A returned list consists of the following fields:

HUM  a list of HUM values for the specified number of analyzed features

seq  a list of vectors, each containing the sequence of class labels
Errors

If there exists NA values for features or class labels no HUM value can be calculated and an error is triggered with message “Values are missing”.

References


See Also

*CalculateHUM_seq*, *CalculateHUM_ROC*

Examples

data(sim)
# Basic example
indexF=c(3,4);
indexClass=2;
allLabel=c("Normal","OrthArthr","OA","Early")
amountL=2
out=CalculateHUM_Ex(sim,indexF,indexClass,allLabel,amountL)

Description

This is the main function of the HUM package. It plots the 2D-ROC curve using the point coordinates, computed by the function *CalculateHUM_ROC*. Optionally visualizes the optimal threshold point, which gives the maximal accuracy of the classifier(feature) (see *CalcROC*).

Usage

*CalculateHUM_Plot*(sel,Sn,Sp,optSn,optSp,HUM,print.optim=TRUE)

Arguments

- `sel` a character value, which is the name of the selected feature.
- `Sn` a numeric vector of the x-coordinates of the ROC curve, which is the specificity values of the standard ROC analysis..
- `Sp` a numeric vector of the y-coordinates of the ROC curve, which is the sensitivity values of the standard ROC analysis..
- `optSn` the optimal specificity value for 2D-ROC construction
- `optSp` the optimal sensitivity value for 2D-ROC construction
- `HUM` a numeric vector of HUM values, calculated using function *CalculateHUM_seq* or *CalculateHUM_Ex*. 
CalculateHUM_ROC

print.optim a boolean parameter to plot the optimal threshold point on the graph. The default value is TRUE.

Details

This function’s main job is to plot the 2D-ROC curve according to the given point coordinates.

Value

The function doesn’t return any value.

Errors

If there exists NA values for specificity or sensitivity values, or HUM values the plotting fails and an error is triggered with message “Values are missing”.

References


See Also

CalculateHUM_seq, CalculateHUM_ROC

Examples

data(sim)
# Basic example
indexF=names(sim[,c(3),drop = FALSE])
indexClass=2
label=unique(sim[,indexClass])
indexLabel=label[1:2]
out=CalculateHUM_seq(sim,indexF,indexClass,indexLabel)
HUMK=out$HUM
seq<out$seq
out=CalculateHUM_ROC(sim,indexF,indexClass,indexLabel,seq)
CalculateHUM_Plot(indexF,out$Sn,out$Sp,out$optSn,out$optSp,HUM)

Description

This is the function of the HUM package for computing the endpoints for ROC curve. It returns a “List” object, consisting of sensitivity and specificity values for 2D-ROC curve and 3D-points for 3D-ROC curve. Also the optimal threshold values are returned.
**Usage**

CalculateHUM_ROC(data,indexF,indexClass,indexLabel,seq)

**Arguments**

data: a dataset, a matrix of feature values for several cases, the additional column with class labels is provided. Class labels could be numerical or character values. The maximal number of classes is ten. The indexClass determines the column with class labels.

indexF: a numeric or character vector, containing the column numbers or column names of the analyzed features.

indexClass: a numeric or character value, containing the column number or column name of the class labels.

indexLabel: a character vector, containing the column names of the class labels, selected for the analysis.

seq: a numeric matrix, containing the permutation of the class labels for all features.

**Details**

This function’s main job is to compute the point coordinates to plot the 2D- or 3D-ROC curve and the optimal threshold values. See the “Value” section to this page for more details. The function calls the calcROC to calculate the point coordinates, optimal thresholds and accuracy of classifier (feature) in the threshold.

Data can be provided in matrix form, where the rows correspond to cases with feature values and class label. The columns contain the values of individual features and the separate column contains class labels. The maximal number of class labels equals 10.

**Value**

The data must be provided without missing values in order to process. A returned list consists of the following fields:

- Sn: a specificity values for 2D-ROC construction and the first coordinate for 3D-ROC construction
- Sp: a sensitivity values for 2D-ROC construction and the second coordinate for 3D-ROC construction
- S3: the third coordinate for 3D-ROC construction
- optSn: the optimal specificity value for 2D-ROC construction and the first coordinate of the optimal threshold for 3D-ROC construction
- optSp: the optimal sensitivity value for 2D-ROC construction and the second coordinate of the optimal threshold for 3D-ROC construction
- optS3: the third coordinate of the optimal threshold for 3D-ROC construction

**Errors**

If there exists NA values for features or class labels no HUM value can be calculated and an error is triggered with message “Values are missing”.
CalculateHUM_seq

References

See Also
CalculateHUM_Ex, CalculateHUM_seq

Examples
```r
data(sim)
# Basic example
indexF=names(sim[,c(3),drop = FALSE])
indexClass=2
label=unique(sim[,indexClass])
indexLabel=label[1:2]
out=CalculateHUM_seq(sim,indexF,indexClass,indexLabel)
HUM=out$HUM
seq=out$seq
out=CalculateHUM_ROC(sim,indexF,indexClass,indexLabel,seq)
```

Description
This is the main function of the HUM package. It computes a HUM value and returns a “List” object, consisting of HUM value and the best permutation of class labels in “seq” vector. This “seq” vector can be passed to the function CalculateHUM_ROC.

Usage
`CalculateHUM_seq(data,indexF,indexClass,indexLabel)`

Arguments
- `data`: a dataset, a matrix of feature values for several cases, the additional column with class labels is provided. Class labels could be numerical or character values. The maximal number of classes is ten. The `indexClass` determines the column with class labels.
- `indexF`: a numeric or character vector, containing the column numbers or column names of the analyzed features.
- `indexClass`: a numeric or character value, containing the column number or column name of the class labels.
- `indexLabel`: a character vector, containing the column names of the class labels, selected for the analysis.
Details

This function’s main job is to compute the maximal HUM value between the all possible permutations of class labels, selected for analysis. See the “Value” section to this page for more details. Before returning, it will call the `CalcGene` function to calculate the HUM value for each feature (object).

Data can be provided in matrix form, where the rows correspond to cases with feature values and class label. The columns contain the values of individual features and the separate column contains class labels. The maximal number of class labels equals 10. The computational efficiency of the function decreases in the case of more than 1000 cases with more than 6 class labels.

Value

The data must be provided without missing values in order to process. A returned list consists of the following fields:

- `hum` a list of HUM values for the specified number of analyzed features
- `seq` a list of vectors, each containing the sequence of class labels

Errors

If there exists NA values for features or class labels no HUM value can be calculated and an error is triggered with message “Values are missing”.

References


See Also

`CalculateHUM_Ex`, `CalculateHUM_ROC`

Examples

```r
data(sim)
# Basic example
indexF=names(sim[,c(3,4)])
indexClass=2
label=unique(sim[,indexClass])
indexLabel=label[1:3]
out=CalculateHUM_seq(sim,indexF,indexClass,indexLabel)
```
dataset

simulated data

Description
This data file consists of six simulated predictors or variables with three class categories. For each class category the values are independently generated from the normal distribution with the mean $\mu_1$, $\mu_2$ and $\mu_3$ and the variances held at unity. The means are varied such that the problems range from near-separable problems, to near-random.

Usage
dataset

Format
A data.frame containing 300 observations of six variables.

Source

See Also
sim

Examples
# load the dataset
data(dataset)

sim
desease data

Description
The data set corresponds to absolute (cells/mm2) or relative (percentage of the cell type in question of the entire inflammatory cell population) densities of 5 major inflammatory cell types in synovial tissue specimens from normal human joints (“Normal”) and from patients with osteoarthritis (“OA”), non-inflammatory orthopedic arthropathies (“Orth.A”), early unclassified arthritis (“EA”), rheumatoid arthritis (“RA”), and chronic septic arthritis (“SeA”). An analysis of this data set with binary and multiclass ROC analysis has been published in Della Beffa PLOS One 2013, which also contains additional details about the data set. The dataset consists of 92 cases with 11 features and disease code.
Usage

sim

Format

A data.frame containing 92 observations of 11 variables.

Source


See Also
dataset

Examples

# load the dataset
data(sim)
# CD15
with(sim, by(CD15,Disease,mean))

# CD20
with(sim,tapply(CD20, Disease, FUN = mean))
with(sim, table(CD20=ifelse(CD20<=mean(CD20), "1", "2"), Disease))
Index

*Topic **auc**
CalcGene, 4  
CalcROC, 6  
Calculate3D, 8  
CalculateHUM_Ex, 10  
CalculateHUM_Plot, 11  
CalculateHUM_ROC, 12  
CalculateHUM_seq, 14  
HUM-package, 2
*Topic **datasets**
dataset, 16  
sim, 16
*Topic **multivariate**
CalcGene, 4  
CalcROC, 6  
Calculate3D, 8  
CalculateHUM_Ex, 10  
CalculateHUM_Plot, 11  
CalculateHUM_ROC, 12  
CalculateHUM_seq, 14  
HUM-package, 2
*Topic **nonparametric**
CalcGene, 4  
CalcROC, 6  
Calculate3D, 8  
CalculateHUM_Ex, 10  
CalculateHUM_Plot, 11  
CalculateHUM_ROC, 12  
CalculateHUM_seq, 14  
HUM-package, 2
*Topic **package**
HUM-package, 2
*Topic **plot**
CalcGene, 4  
CalcROC, 6  
Calculate3D, 8  
CalculateHUM_Ex, 10  
CalculateHUM_Plot, 11  
CalculateHUM_ROC, 12  
CalculateHUM_seq, 14  
HUM-package, 2

CalculateHUM_seq, 14  
HUM-package, 2

*Topic **roc**
CalcGene, 4  
CalcROC, 6  
Calculate3D, 8  
CalculateHUM_Ex, 10  
CalculateHUM_Plot, 11  
CalculateHUM_ROC, 12  
CalculateHUM_seq, 14  
HUM-package, 2

CalcGene, 2, 3, 4, 10, 15  
CalcROC, 2, 3, 6, 8, 11, 13  
Calculate3D, 3, 8  
CalculateHUM_Ex, 2, 5, 7, 10, 11, 14, 15  
CalculateHUM_Plot, 3, 11  
CalculateHUM_ROC, 2–5, 7–12, 12, 14, 15  
CalculateHUM_seq, 2, 9, 11, 12, 14, 14  

dataset, 16, 17

HUM (HUM-package), 2
HUM-package, 2

sim, 16, 16