

# Package: quantileDA (via r-universe)

September 16, 2024

**Type** Package  
**Title** Quantile Classifier  
**Version** 1.2  
**Date** 2024-03-19  
**Author** Marco Berrettini, Christian Hennig, Cinzia Viroli  
**Maintainer** Cinzia Viroli <cinzia.viroli@unibo.it>  
**Description** Code for centroid, median and quantile classifiers.  
**License** GPL-3  
**NeedsCompilation** no  
**Date/Publication** 2024-03-19 17:00:02 UTC  
**Repository** <https://viroli.r-universe.dev>  
**RemoteUrl** <https://github.com/cran/quantileDA>  
**RemoteRef** HEAD  
**RemoteSha** 703f86867f68fbedf0d61bb8aba89f04cc15658d

## Contents

ais . . . . .	2
balanced.folds . . . . .	3
centroidcl . . . . .	3
fn_vw . . . . .	4
galtonskew . . . . .	4
init.theta . . . . .	5
kelleykew . . . . .	5
mapClass . . . . .	6
misc . . . . .	6
permute.rows . . . . .	6
plot.quantileDA . . . . .	7
print.quantileDA . . . . .	7
quantilecl . . . . .	7
quantilecl.vw . . . . .	9

quantileCV . . . . .	10
skewness . . . . .	11
theta.cl . . . . .	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

ais	<i>Australian Institute of Sport data</i>
-----	---

---

## Description

Data on 102 male and 100 female athletes collected at the Australian Institute of Sport, courtesy of Richard Telford and Ross Cunningham.

## Usage

```
data(ais)
```

## Format

A data frame with 202 observations on the following 13 variables.

sex A factor with levels female male  
 sport A factor with levels B\_Ball Field Gym Netball Row Swim T\_400m T\_Sprnt Tennis W\_Polo  
 rcc A numeric vector: red cell count  
 wcc A numeric vector: white cell count  
 Hc A numeric vector: Hematocrit  
 Hg A numeric vector: Hemoglobin  
 Fe A numeric vector: plasma ferritin concentration  
 bmi A numeric vector: body mass index  
 ssf A numeric vector: sum of skin folds  
 Bfat A numeric vector: body fat percentage  
 lbm A numeric vector: lean body mass  
 Ht A numeric vector: height (cm)  
 Wt A numeric vector: weight (kg)

## Source

Cook and Weisberg (1994), *An Introduction to Regression Graphics*. John Wiley & Sons, New York.

## Examples

```
data(ais)
attach(ais)
pairs(ais[,c(3:4,10:13)], main = "AIS data")
plot(Wt~sport)
```

---

balanced.folds	<i>Internal function used in the cross-validation of the quantile classifier</i>
----------------	--

---

### Description

Internal function used the cross-validation of the quantile classifier

---

centroidcl	<i>A function that performs the centroid classifier</i>
------------	---

---

### Description

Given a training and a test set, the function apply the centroid classifier and returns the classification labels of the observations in the training and in test set. It also gives the training misclassification rate and the test misclassification rate, if the truth class labels of the test set are provided in input.

### Usage

```
centroidcl(train, test, cl, cl.test = NULL)
```

### Arguments

train	A matrix of data (the training set) with observations in rows and variables in column. It can be a matrix or a dataframe.
test	A matrix of data (the test set) with observations in rows and variables in columns. It can be a matrix or a dataframe.
cl	A vector of class labels for each sample of the training set. It can be factor or numerical.
cl.test	A vector of class labels for each sample of the test set (optional)

### Details

centroidcl carries out the centroid classifier and predicts classification.

### Value

A list with components

cl.train	Predicted classification in the training set
cl.test	Predicted classification in the test set
me.train	Misclassification error in the training set
me.test	Misclassification error in the test set (only if cl.test is available)

**Author(s)**

Christian Hennig, Cinzia Viroli

**See Also**

See Also [theta.cl](#)

**Examples**

```
data(ais)
x=ais[,3:13]
cl=as.double(ais[,1])
set.seed(22)
index=sample(1:202,152,replace=FALSE)
train=x[index,]
test=x[-index,]
cl.train=cl[index]
cl.test=cl[-index]
out.c=centroidcl(train,test,cl.train,cl.test)
out.c$me.test
misc(out.c$cl.test,cl.test)
```

---

fn\_vw

*Internal function for the quantile classifier with variable-wise thetas*

---

**Description**

Internal function for the quantile classifier with variable-wise thetas

---

galtonskew

*A function that compute the Galton's skewness*

---

**Description**

The function compute the Galton's skewness index on a set of observations.

**Usage**

```
galtonskew(x)
```

**Arguments**

x                    A vector of observations.

**Value**

A scalar which measures the Galton's skewness

**Author(s)**

Christian Hennig, Cinzia Viroli

**See Also**

See Also [kelleyskew](#)

**Examples**

```
data(ais)
galtonskew(ais[,4])
```

---

<code>init.theta</code>	<i>Internal function for the quantile classifier</i>
-------------------------	--

---

**Description**

Internal function for the quantile classifier

---

<code>kelleyskew</code>	<i>A function that compute the Kelley's skewness</i>
-------------------------	--

---

**Description**

The function compute the Kelley's skewness index on a set of observations.

**Usage**

```
kelleyskew(x)
```

**Arguments**

`x`                    A vector of observations.

**Value**

A scalar which measures the Kelley's skewness

**Author(s)**

Christian Hennig, Cinzia Viroli

**See Also**

See Also [galtonskew](#)

**Examples**

```
data(ais)
kelley skew(ais[,4])
```

---

mapClass	<i>Internal function for the quantile classifier</i>
----------	--

---

**Description**

Internal function for the quantile classifier

---

misc	<i>Misclassification error</i>
------	--------------------------------

---

**Description**

An internal function which computes the misclassification error between two partitions

**Usage**

```
misc(classification, truth)
```

**Arguments**

`classification` A numeric or character vector of class labels.

`truth` A numeric or character vector of truth class labels. The length of truth should be the same as that of classification.

**Value**

The misclassification error (a scalar).

---

permute.rows	<i>Internal function used by the quantile classifier</i>
--------------	--

---

**Description**

Internal function used by the quantile classifier

---

plot.quantileDA	<i>Internal function for plotting the results of the quantile classifier</i>
-----------------	--

---

**Description**

Internal function for plotting the results of the quantile classifier

---

print.quantileDA	<i>Internal function for printing the results of the quantile classifier</i>
------------------	--

---

**Description**

Internal function for printing the results of the quantile classifier

---

quantilecl	<i>A function that applies the quantile classifier for a given set of quantile probabilities and selects the best quantile classifier in the training set.</i>
------------	--

---

**Description**

The function applies the quantile classifier for a set of quantile probabilities and selects the optimal probability that minimize the misclassification rate in the training set.

**Usage**

```
quantilecl(train, test, cl, theta = NULL,
           cl.test = NULL, skew.correct="Galton")
```

**Arguments**

train	A matrix of data (the training set) with observations in rows and variables in columns. It can be a matrix or a dataframe.
test	A matrix of data (the test set) with observations in rows and variables in columns. It can be a matrix or a dataframe.
cl	A vector of class labels for each sample of the training set. It can be factor or numerical.
theta	A vector of quantile probabilities (optional)
cl.test	If available, a vector of class labels for each sample of the test set (optional)
skew.correct	Skewness measures applied to correct the skewness direction of the variables. The possible choices are: Galton's skewness (default), Kelley's skewness and the conventional skewness index based on the third standardized moment

**Details**

quantile\_cl carries out the quantile classifier for a set of quantile probabilities and selects the optimal probability that minimize the misclassification rate in the training set. The values of the quantile probabilities can be given in input or automatically selected in a equispaced range of 49 values between 0 and 1. The data in the training and test samples are preprocessed so that the variables used for the quantile estimator all have the same (positive) direction of skewness according to different measures of skewness: Galton's skewness, Kelley's skewness or conventional skewness index.

**Value**

A list with components

train.rates	Misclassification errors for each quantile probability in the training set
test.rates	Misclassification errors for each quantile probability in the test set
thetas	The list of optimal quantile probabilities for each variable
theta.choice	The quantile probability that gives the less misclassification error in the training set
me.train	Misclassification error in the training set
me.test	Misclassification error in the test set (only if cl.test is available)
train	The matrix of data (training set) with observations in rows and variables in columns
test	The matrix of data (test set) with observations in rows and variables in columns
cl.train	Predicted classification in the training set
cl.test	Predicted classification in the test set
cl.train.0	The true classification labels in the training set
cl.test.0	The true classification labels in the test set (if available)

**Author(s)**

Christian Hennig, Cinzia Viroli

**See Also**

See Also [quantilecl.vw](#)

**Examples**

```
data(ais)
x=ais[,3:13]
cl=as.double(ais[,1])
set.seed(22)
index=sample(1:202,152,replace=FALSE)
train=x[index,]
test=x[-index,]
cl.train=cl[index]
```



```

cl.test=cl[-index]
out.q=quantilecl(train,test,cl.train,cl.test=cl.test)
out.q$me.test
print(out.q)
plot(out.q)

```

---

quantilecl.vw      *A function to apply the quantile classifier that uses a different optimal quantile probability for each variable*

---

### Description

A function to apply the quantile classifier that uses a different optimal quantile probability for each variable

### Usage

```
quantilecl.vw(train, test, cl, theta = NULL, cl.test = NULL)
```

### Arguments

train	A matrix of data (the training set) with observations in rows and variables in columns. It can be a matrix or a dataframe.
test	A matrix of data (the test set) with observations in rows and variables in columns. It can be a matrix or a dataframe.
cl	A vector of class labels for each sample of the training set. It can be factor or numerical.
theta	Given $p$ variables, a vector of length $p$ of quantile probabilities (optional)
cl.test	If available, a vector of class labels for each sample of the test set (optional)

### Details

quantilecl.vw carries out the quantile classifier by using a different optimal quantile probability for each variable selected in the training set.

### Value

A list with components

vseq	The value of the objective function at each iteration
thetas	The vector of quantile probabilities
me.train	Misclassification error for the best quantile probability in the training set
me.test	Misclassification error for the best quantile probability in the test set (only if cl.test is available)
cl.train	Predicted classification in the training set
cl.test	Predicted classification in the test set
lambda	The vector of estimated scale parameters

**Author(s)**

Marco Berrettini, Christian Hennig, Cinzia Viroli

**See Also**

See Also [quantilecl](#)

**Examples**

```
data(ais)
x=ais[,3:7]
cl=as.double(ais[,1])
set.seed(22)
index=sample(1:202,152,replace=FALSE)
train=x[index,]
test=x[-index,]
cl.train=cl[index]
cl.test=cl[-index]
out.q=quantilecl.vw(train,test,cl.train,cl.test=cl.test)
out.q$me.test
```

---

quantileCV

*A function to cross-validate the quantile classifier*

---

**Description**

Balanced cross-validation for the quantile classifier

**Usage**

```
quantileCV(x, cl, nfold = min(table(cl)),
  folds = balanced.folds(cl, nfold), theta=NULL, seed = 1, varying = FALSE)
```

**Arguments**

x	A matrix of data (the training set) with observations in rows and variables in columns (it can be a matrix or a dataframe)
cl	A vector of class labels for each sample (factor or numerical)
nfold	Number of cross-validation folds. Default is the smallest class size. Admitted values are from 1 to the smallest class size as maximum fold number.
folds	A list with nfold components, each component a vector of indices of the samples in that fold. By default a (random) balanced cross-validation is used
theta	A vector of quantile probabilities (optional)
seed	Fix the seed of the running. Default is 1
varying	If TRUE a different quantile for each variable is selected in the training set. If FALSE (default) an unique quantile is used.

**Details**

quantileCV carries out cross-validation for a quantile classifier.

**Value**

A list with components

test.rates	Mean of misclassification errors in the cross-validation test sets for each quantile probability (available if varying is FALSE)
train.rates	Mean of misclassification errors in the cross-validation train sets for each quantile probability (available if varying is FALSE)
thetas	The fitted quantile probabilities
theta.choice	Value of the chosen quantile probability in the training set
me.test	Misclassification errors in the cross validation test sets for the best quantile probability
me.train	Misclassification errors in the cross validation training sets for the best quantile probability
me.median	Misclassification errors in the cross validation test sets of the median classifier
me.centroid	Misclassification errors in the cross validation test sets of the centroid classifier
folds	The cross-validation folds used

**Author(s)**

Christian Hennig, Cinzia Viroli

**Examples**

```
data(ais)
x=ais[,3:13]
cl=as.double(ais[,1])
out=quantileCV(x,cl,nfold=2)
```

---

skewness

*A function that compute the conventional skewness measure*

---

**Description**

A function that compute the conventional skewness measure according to the third standardized moment of x

**Usage**

```
skewness(x)
```

**Arguments**

x                    A vector of observations.

**Value**

A scalar which measures the skewness

**Author(s)**

Christian Hennig, Cinzia Viroli

**See Also**

See Also [galtonskew](#)

**Examples**

```
data(ais)
skewness(ais[,4])
```

---

theta.cl	<i>A function to perform the quantile classifier for a given quantile probability</i>
----------	---

---

**Description**

Given a certain quantile probability, the function compute the quantile classifier on the training set and gives the predicted class labels in the training and test set. It also computes the training misclassification rate and the test misclassification rate, when the truth labels of the test set are available. When the quantile probability is 0.5 the function compute the median classifier.

**Usage**

```
theta.cl(train, test, cl, theta, cl.test = NULL)
```

**Arguments**

train	A matrix of data (the training set) with observations in rows and variables in columns. It can be a matrix or a dataframe.
test	A matrix of data (the test set) with observations in rows and variables in columns. It can be a matrix or a dataframe.
cl	A vector of class labels for each sample of the training set. It can be factor or numerical.
theta	The quantile probability. If 0.5 the median classifier is applied
cl.test	If available, a vector of class labels for each sample of the test set (optional)

**Details**

theta.cl carries out quantile classifier for a given quantile probability.

**Value**

A list with components

cl.train	Predicted classification in the training set
cl.test	Predicted classification in the test set
me.train	Misclassification error in the training set
me.test	Misclassification error in the test set (only if cl.test is available)

**Author(s)**

Christian Hennig, Cinzia Viroli

**See Also**

See Also [centroidcl](#)

**Examples**

```
data(ais)
x=ais[,3:13]
cl=as.double(ais[,1])
set.seed(22)
index=sample(1:202,152,replace=FALSE)
train=x[index,]
test=x[-index,]
cl.train=cl[index]
cl.test=cl[-index]
out.m=theta.cl(train,test,cl.train,0.5,cl.test)
out.m$me.test
misc(out.m$cl.test,cl.test)
```

# Index

## \* datasets

ais, 2

## \* multivariate

centroidcl, 3

galtonskew, 4

kelleyskew, 5

misc, 6

quantilecl, 7

quantilecl.vw, 9

quantileCV, 10

skewness, 11

theta.cl, 12

ais, 2

balanced.folds, 3

centroidcl, 3, 13

fn\_vw, 4

galtonskew, 4, 5, 12

init.theta, 5

kelleyskew, 5, 5

mapClass, 6

misc, 6

permute.rows, 6

plot.quantileDA, 7

print.quantileDA, 7

quantilecl, 7, 10

quantilecl.vw, 8, 9

quantileCV, 10

skewness, 11

theta.cl, 4, 12