

Package ‘freestats’

February 19, 2015

Type Package

Title Statistical algorithms used in common data mining course

Version 0.0.3

Date 2014-05-06

Author Xiaoyao Yang

Maintainer Xiaoyao Yang <hyper8866@gmail.com>

Description A collections of useful statistical functions used in Columbia course W4240/W4400.

License BSD_3_clause + file LICENSE

LazyLoad yes

ByteCompile TRUE

Imports MASS, mvtnorm

NeedsCompilation no

Repository CRAN

Date/Publication 2014-05-07 08:02:00

R topics documented:

adaBoost	2
classify	3
classify.ab	4
classify.default	5
classify.ds	5
classify.pt	6
cost.01	7
cost.mse	8
cv.work	8
decisionStump	9
fakedata	10
freestats	10
perceptrain	11

Index	12
--------------	-----------

adaBoost

Adaboost algorithm

Description

Do classification using adaboost algorithm with decisionStump as weak learner

Usage

```
adaBoost(train = decisionStump, dat.train, y.train, B = 10, ...)
```

Arguments

train	Function of weak learner that would be used in adaboost, must have form train(dat.train,w,y.train)
dat.train	Training data set
y.train	Label for training data set
B	Number of weak learners that will used
...	Other parameters that need to passed in train function

Details

Train function can be any weak learner algorithm. For now, train function must has form train(X,w,y,...). see more in [decisionStump](#)

If you have any good weak learner but can't use it in this function, feel free to let me know.

Value

alpha	The weight for different weak learners
allPars	A list of parameters for different weak learners

Author(s)

Xiaoyao Yang

Examples

```
set.seed(1024)
z <- runif(n=5)
mydata <- fakedata(w=z,n=100)
X<- mydata$S[,1:4]
y <- mydata$y
res <- adaBoost(dat.train=X,y.train=y,B=3)
```

classify	<i>Generic classify function</i>
----------	----------------------------------

Description

Returning classification result

Usage

```
classify(pars, dat)
```

Arguments

pars	Result returned by training algorithm
dat	Data set to be classified

Details

Returning label for classification problem

This is the S3 generic method for returning classification result.

For more information [classify.default](#)

Value

An label obtained by classification method determined by pars

Author(s)

Xiaoyao Yang

Examples

```
set.seed(1024)
z <- runif(n=5)
mydata <- fakedata(w=z,n=100)
X<- mydata$S[,1:4]
y <- mydata$y
w <- rep(1/100,100)
pars <- decisionStump(X=X,w=w,y=y)
classify(pars,X)
```

`classify.ab`*Classify for adaboost result*

Description

Returning classification result returned by function `adaBoost`

Usage

```
## S3 method for class 'ab'  
classify(pars, dat)
```

Arguments

<code>pars</code>	Result returned by adaboost algorithm
<code>dat</code>	Data set to be classified

Details

Specific designed for result returned by `adaBoost` function

For more information [adaBoost](#)

Value

An label (vector) obtained by classification method determined by `pars`

Author(s)

Xiaoyao Yang

Examples

```
set.seed(1024)  
z <- runif(n=5)  
mydata <- fakedata(w=z,n=100)  
X<- mydata$S[,1:4]  
y <- mydata$y  
res <- adaBoost(dat.train=X,y.train=y,B=3)  
classify.ab(res,X)
```

classify.default	<i>Default method for classify</i>
------------------	------------------------------------

Description

Returning classification result

Usage

```
## Default S3 method:  
classify(pars, dat)
```

Arguments

pars	Result returned by training algorithm
dat	Data set to be classified

Value

An label obtained by classification method determined by pars

Author(s)

Xiaoyao Yang

Examples

```
set.seed(1024)  
z <- runif(n=5)  
mydata <- fakedata(w=z,n=100)  
X<- mydata$S[,1:4]  
y <- mydata$y  
w <- rep(1/100,100)  
pars <- decisionStump(X=X,w=w,y=y)  
classify.default(pars,X)
```

classify.ds	<i>classify.ds</i>
-------------	--------------------

Description

Returning classification for DecisionStump

Usage

```
## S3 method for class 'ds'  
classify(...)
```

Arguments

... See [classify.default](#) for argument details

Details

Returning label for classification problem

[classify](#) is the S3 generic method for returning classification result.

For more information [classify.default](#)

Value

See [classify.default](#) for more information.

Author(s)

Xiaoyao Yang

Examples

```
set.seed(1024)
z <- runif(n=5)
mydata <- fakedata(w=z, n=100)
X<- mydata$S[,1:4]
y <- mydata$y
w <- rep(1/100,100)
pars <- decisionStump(X=X,w=w,y=y)
classify.ds(pars,X)
```

classify.pt

Classify for perceptrain result

Description

Returning classification result

Usage

```
## S3 method for class 'pt'
classify(pars, dat)
```

Arguments

pars Result returned by training algorithm
dat Data set to be classified (Here cbind(dat,1)=S)

Details

For consistency, using X(data set) instead of S ([dataset,1]) for classify function.

For more information [perceptrain](#)

Value

An label obtained by classification method determined by pars

Author(s)

Xiaoyao Yang

Examples

```
set.seed(1024)
z <- runif(n=3)
mydata <- fakedata(w=z,n=100)
r <- perceptrain(S=mydata$S,y=mydata$y,alpha_k=1,endcost=0)
classify.pt(r,mydata$S[,1:(NCOL(mydata$S)-1)])
```

cost.01

cost.01

Description

0-1 loss function

Usage

```
cost.01(y, yhat)
```

Arguments

y	Response
yhat	The predicted value

Value

0-1 loss

cost.mse	<i>cost.mse</i>
----------	-----------------

Description

mse

Usage

cost.mse(y, yhat)

Arguments

y	Response
yhat	The predicted value

Value

Mean Square Error

cv.work	<i>K-fold cross validation</i>
---------	--------------------------------

Description

Calculate cross-validation error

Usage

cv.work(fun, k = 5, data, cost, response = "y", ...)

Arguments

fun	The model function to call on the data
k	The number of folds
data	The data
cost	Cost function for the error: 'cost.mse', 'cost.01'
response	Character vector indicating which column is the response
...	Extra arguments for model function

Value

A single cross-validated error

Author(s)

Xiaoyao Yang

Examples

```

set.seed(188)
X <- rnorm(n=100, mean=3, sd=2)
y <- rnorm(100) + X
dat <- data.frame(y=y, X=X)
cv.work(fun=lm, k=5, data=dat, cost=cost.mse, response='y', formula=y~X)

```

decisionStump

Decision Stump Algorithm

Description

Do classification with tree method in one step

Usage

```
decisionStump(X, w, y)
```

Arguments

X	Data matrix / Data frame
w	Weight that given to each observation. Used in calculate cost function.
y	Class label for data points in X, must be -1 or 1

Value

j	The best dimation to cut the tree
theta	Value that seperate tree in the best dimation
m	the routine label value (for now only 1)

Author(s)

Xiaoyao Yang

Examples

```

set.seed(1024)
z <- runif(n=5)
mydata <- fakedata(w=z, n=100)
X<- mydata$S[,1:4]
y <- mydata$y
w <- rep(1/100, 100)
pars <- decisionStump(X=X, w=w, y=y)

```

fakedata

Generating seperatable data

Description

fakedata for algorithm like perceptron

Usage

```
fakedata(w, n)
```

Arguments

w	The $w[1:d]$ is the normal vector of a hyperplane, $w[d+1] = -c$ is the negative offset parameter
n	Sample size

Value

S	n by (d+1) sample matrix with last col equal to 1
y	Normal vector of the associated class labels

Author(s)

Xiaoyao Yang

Examples

```
fakedata(c(1,1,1),10)
```

freestats*Helper functions*

Description

A useful statistical functions specifically for W4240/W4400

perceptrain *An original perceptron algorithm*

Description

Train data with perceptron algorithm

Usage

```
perceptrain(S, y, alpha_k = 1, endcost = 0)
```

Arguments

S	Each row represents a data points with last column equal to 1; S=[X,1]
y	Class label for data points in S
alpha_k	The speed of converge
endcost	The termination condition of cost function

Details

S is especially designed for perceptron.

For more information [fakedata](#)

Value

z	Normal vector of a hyperplane
Z_history	Trajectory of normal vector of a hyperplane
NumofIteration	Number of iterations for algorithm

Author(s)

Xiaoyao Yang

Examples

```
set.seed(1024)
z <- runif(n=3)
mydata <- fakedata(w=z,n=100)
r <- perceptrain(S=mydata$S,y=mydata$y,alpha_k=1,endcost=0)
r
```

Index

`adaBoost`, [2](#), [4](#)

`classify`, [3](#), [6](#)

`classify.ab`, [4](#)

`classify.default`, [3](#), [5](#), [6](#)

`classify.ds`, [5](#)

`classify.pt`, [6](#)

`cost.01`, [7](#)

`cost.mse`, [8](#)

`cv.work`, [8](#)

`decisionStump`, [2](#), [9](#)

`fakedata`, [10](#), [11](#)

`freestat (freestats)`, [10](#)

`freestats`, [10](#)

`freestats-package (freestats)`, [10](#)

`perceptrain`, [7](#), [11](#)