

Description

The data are divided into v non-overlapping subsets of roughly equal size. Then, feature selection is applied on $(v-1)$ of the subsets, which are also used to fit the LogitBoost classifier. Then, predictions are made for the left out subsets, and the process is repeated for each of the v subsets.

Usage

```
crossval(x, y, v=length(y), mfinal=100, presel=0, estimate=0, verbose=F)
```

Arguments

<code>x</code>	A matrix with n rows (different individuals) and p columns (different genes) containing expression values.
<code>y</code>	A vector of length n containing the class labels from individuals of K different classes. The labels need to be coded by consecutive integers from 0 to $(K-1)$.
<code>v</code>	An integer, specifying the type of v -fold cross validation. The default, $v=length(y)$ means leave-one-out cross validation. Besides this, every value between 2 and $length(y)$ is valid and means that roughly every v -th observation is left out. Make sure that (especially for multiclass problems) this is a sensible partition into training and test data.
<code>mfinal</code>	An integer, describing the number of iterations for which boosting should be run. The default value is $mfinal=100$, which is a reasonable choice for gene expression data.
<code>presel</code>	An integer, giving the number of features to be used for classification. If $presel=0$, no feature preselection is carried out.
<code>estimate</code>	An integer, specifying the v of an additional, internal v -fold cross validation on the respective training data for stopping parameter estimation. Please note that this is (especially for larger values of 'estimate') extremely time consuming. The default value of $estimate=0$ means no stopping parameter estimation.
<code>verbose</code>	Logical, indicates whether comments should be given.

Details

The computation of the stopping parameter estimate is computationally very expensive and time consuming.

Value

<code>probs</code>	Array, whose rows contain out of sample probabilities that the class labels are predicted as 1, for every boosting iteration. For multiclass problems, the third dimension of the array are the probabilities for the K binary one-against-all partitions of the data.
<code>loglikeli</code>	Array, contains the log-likelihood across the training instances for determination of the stopping parameter if <code>estimate>0</code> . For multiclass problems, the third dimension of the array contains the values for the K binary one-against-all partitions of the data.

Author(s)

Marcel Dettling

References

See "Boosting for Tumor Classification of Gene Expression Data", Dettling and Buhlmann (2002), available on the web page <http://stat.ethz.ch/~dettling/boosting.html>

See Also

`logitboost`, `summarize`

Examples

```
data(leukemia)

## An example without stopping parameter estimation
fit <- crossval(leukemia.x,leukemia.y,v=5,mfinal=100,presel=75,verbose=TRUE)
summarize(fit,leukemia.y)

## 4-fold cross validation with stopping estimation by 3-fold-cv
fit <- crossval(leukemia.x,leukemia.y,v=4,presel=50,estimate=3,verbose=TRUE)
summarize(fit,leukemia.y)
```

`cv.binary`

These are internal functions for cross validation with LogitBoost

Description

Not to be called by the user.

Author(s)

Marcel Dettling

References

"Boosting for Tumor Classification with Gene Expression Data", see "<http://stat.ethz.ch/det-tling/boosting.html>"

See Also

crossval

leukemia

A part of the famous AML/ALL-leukemia dataset

Description

This is the training set of the famous AML/ALL-leukemia dataset from the Whitehead Institute. It has been reduced to 250 genes, about the half of which are very informative for classification, whereas the other half was chosen randomly.

Usage

```
data(leukemia)
```

Format

Contains three R-objects: The expression matrix leukemia.x, the associated binary response variable leukemia.y, and the associated 3-class response variable leukemia.z

Source

<http://www.genome.wi.mit.edu/MPR>

References

First published in Golub et al: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science 1999, 286: 531-538.

Examples

```
data(leukemia)
str(leukemia.x)
str(leukemia.y)
str(leukemia.z)
par(mfrow=c(1,2))
plot(leukemia.x[,56], leukemia.y)
plot(leukemia.x[,174], leukemia.z)
```

Description

An implementation of the LogitBoost classification algorithm with decision stumps as weak learners. Additionally, a feature preselection method for handling datasets with many explanatory variables and estimation of the stopping parameter via v -fold cross validation are provided.

Usage

```
logitboost(xlearn, ylearn, xtest, mfinal, presel = 0, estimate = 0,  
verbose = FALSE)
```

Arguments

<code>xlearn</code>	A matrix, whose n rows contain the training instances.
<code>ylearn</code>	A vector of length n containing the class labels from individuals of K different classes. The labels need to be coded by consecutive integers from 0 to $(K-1)$.
<code>xtest</code>	A matrix, whose rows contain the test instances.
<code>mfinal</code>	An integer, describing the number of iterations for which boosting should be run.
<code>presel</code>	An integer, giving the number of features to be used for classification. If <code>presel=0</code> , no feature preselection is carried out.
<code>estimate</code>	An integer, specifying the v of an additional, internal v -fold cross validation on the respective training data for stopping parameter estimation. Please note that this is (especially for larger values of 'estimate') extremely time consuming. The default value of <code>estimate=0</code> means no stopping parameter estimation.
<code>verbose</code>	Logical, indicates whether comments should be given.

Value

<code>probs</code>	Array, whose rows contain out of sample probabilities that the class labels are predicted as 1, for every boosting iteration. For multiclass problems, the third dimension of the array are the probabilities for the K binary one-against-all partitions of the data.
<code>loglikeli</code>	Array, contains the log-likelihood across the training instances for determination of the stopping parameter if <code>estimate>0</code> . For multiclass problems, the third dimension of the array contains the values for the K binary one-against-all partitions of the data.

Author(s)

Marcel Dettling

References

See "Boosting for Tumor Classification of Gene Expression Data", Dettling and Buhlmann (2002), available on the web page <http://stat.ethz.ch/dettling/boosting.html>

See Also

`crossval`, `summarize`

Examples

```
data(leukemia)

## Dividing the leukemia dataset into training and test data
xlearn <- leukemia.x[c(1:20, 34:38),]
ylearn <- leukemia.y[c(1:20, 34:38)]
xtest  <- leukemia.x[21:33,]
ytest  <- leukemia.y[21:33]

## An example without stopping parameter estimation
fit <- logitboost(xlearn, ylearn, xtest, mfinal=100, pre=75, verbose=TRUE)
summarize(fit, ytest)

## Now with stopping parameter estimation by 4-fold cross validation
fit <- logitboost(xlearn, ylearn, xtest, mfinal=100, pre=75, esti=4, verb=TRUE)
summarize(fit, ytest)
```

`score`

Computes the score function of gene expression vectors

Description

The score function measures how well an explanatory variable discriminates a given binary response. It can be interpreted as counting for each observation having response zero, the number of individuals of response class one that have smaller expression values, and summing up these quantities and is equivalent to the test statistic of the Wilcoxon test.

Usage

```
score(x, resp)
```

Arguments

`x` A numerical vector, containing the value of the explanatory variable for all instances.

`resp` Vector, containing the class labels of the instances which have to be coded by 0 and 1.

Value

An integer, the score of that particular explanatory variable.

Author(s)

Marcel Dettling

References

See "Boosting for Tumor Classification with Gene Expression Data", Dettling and Buhlmann (2002), available on the web page <http://stat.ethz.ch/dettling/boosting.html>

Examples

```
data(leukemia)

plot(leukemia.x[,69],leukemia.y)
title(paste("Score = ", score(leukemia.x[,69], leukemia.y)))
```

<code>summarize</code>	<i>Summarizes the output of <code>crossval()</code> and <code>logitboost()</code> by printing and plotting</i>
------------------------	--

Description

Prints and plots error-rates for optimal, fixed and (optionally) estimated stopping times when predicting a test set with `logitboost()`, or when running v-fold cross validation via `crossval()`.

Usage

```
summarize(boost.out, resp, mout=100, grafik=T)
```

Arguments

<code>boost.out</code>	A list, obtained as output of either <code>logitboost()</code> or <code>crossval()</code>
<code>resp</code>	A numerical vector, containing the true response labels of the K classes as consecutive integers from 0 to (K-1).
<code>mout</code>	An integer, giving the number of iterations of the boosting procedure, for which the error rate should be printed. The default value <code>mout=100</code> is usually a good choice for gene expression data, which can well be inspected visually by the boosting error curve.
<code>grafik</code>	Logical flag, indicates whether the boosting error curve should be plotted or not. The default is TRUE.

Value

Prints and plots the error-rates from the LogitBoost procedure.

Author(s)

Marcel Dettling

References

See "Boosting for Tumor Classification with Gene Expression Data", Dettling and Buhlmann (2002), available on the web page <http://stat.ethz.ch/dettling/boosting.html>

Examples

```
data(leukemia)

## An example without stopping parameter estimation
fit <- crossval(leukemia.x, leukemia.y, v=5, mfinal=100, presel=75, verbose=TRUE)
summarize(fit, leukemia.y, grafik=FALSE)
summarize(fit, leukemia.y, mout=57)
```