**crossval**  
*Runs v-fold cross validation with LogitBoost*

**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**

- **x**  
  A matrix with n rows (different individuals) and p columns (different genes) containing expression values.

- **y**  
  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).

- **v**  
  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.

- **mfinal**  
  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.

- **presel**  
  An integer, giving the number of features to be used for classification. If presel=0, no feature preselection is carried out.

- **estimate**  
  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.

- **verbose**  
  Logical, indicates whether comments should be given.

**Details**

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

probs

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 probabilites for the K binary one-against-all partitions of the data.

loglikelihood

Array, contains the log-likelihood across the training instances for determination of the stopping parameter if estimate>0. 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 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/dettling/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

```r
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


Examples

```r
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)
```
**logitboost**        *LogitBoost*

**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 and estimation of the stopping parameter via v-fold cross validation are provided.

**Usage**

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

**Arguments**

- `xlearn` A matrix, whose n rows contain the training instances.
- `ylearn` 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).
- `xtest` A matrix, whose rows contain the test instances.
- `mfinal` An integer, describing the number of iterations for which boosting should be run.
- `presel` An integer, giving the number of features to be used for classification. If presel=0, no feature preselection is carried out.
- `estimate` 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.
- `verbose` Logical, indicates whether comments should be given.

**Value**

- `probs` 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.
- `loglikeli` Array, contains the log-likelihood across the training instances for determination of the stopping parameter if estimate>0. 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 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, presel=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 coded by 0 and 1.
Value

An integer, the score of that particular explanatory variable.

Author(s)

Marcel Dettling

References


Examples

data(leukemia)

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

summarize Summarizes the output of crossval() and logitboost() by printing
and plotting

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

boost.out A list, obtained as output of either logitboost() or crossval()
resp A numerical vector, containing the true response labels of the K classes
as consecutive integers from 0 to (K-1).
mout An integer, giving the number of iterations of the boosting procedure, for
which the error rate should be printed. The default value mout=100 is
usually a good choice for gene expression data, which can well be inspected
visually by the boosting error curve.
grafik 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


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)