

The boost Package

December 10, 2004

Title Boosting Methods for Real and Simulated Data

Version 1.0-0

Author Marcel Dettling

Description Contains a collection of boosting methods, these are 'BagBoost', 'LogitBoost', 'AdaBoost' and 'L2Boost',

Maintainer Marcel Dettling, <dettling@jhu.edu>

URL <http://stat.ethz.ch/~dettling>

License GPL 2.0

R topics documented:

adaboost	1
bagboost	3
l2boost	4
learner	5
leukemia	5
logitboost	6
simulator	7
summarize	9

Index	11
--------------	-----------

adaboost	<i>adaboost</i>
----------	-----------------

Description

An implementation of the AdaBoost algorithm for binary classification

Usage

```
adaboost(xlearn, ylearn, xtest, presel = 200, mfinal = 100)
```

Arguments

<code>xlearn</code>	A (n x p)-matrix, where rows correspond to training instances and columns contain the predictor variables.
<code>ylearn</code>	A vector of length n containing the class labels, which need to be coded by 0 and 1.
<code>xtest</code>	A (m x p)-matrix, where rows correspond to test instances and columns contain the predictor variables.
<code>presel</code>	An integer, giving the number of features to be pre-selected according to the Wilcoxon test statistic. Default is <code>presel=200</code> features. If <code>presel=0</code> , no feature preselection is carried out.
<code>mfinal</code>	An integer, the number of iterations for which boosting is run. Defaults to <code>mfinal=100</code> iterations.

Value

The function outputs an array, whose rows contain out-of-sample probabilities that the class labels are predicted as being of class 1, for every boosting iteration.

Author(s)

Marcel Dettling

References

- o "Boosting for Tumor Classification with Gene Expression Data", Marcel Dettling and Peter Bühlmann. *Bioinformatics* (2003), Vol. 19, p. 1061–1069.
- o "BagBoosting for Tumor Classification with Gene Expression Data", Marcel Dettling. To appear in *Bioinformatics* (2005).
- o Further information is available from the webpage <http://stat.ethz.ch/~dettling>

Examples

```
data(leukemia, package = "boost")

## 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]

## Classification with adaboost
fit <- adaboost(xlearn, ylearn, xtest, presel=50, mfinal=20)
summarize(fit, ytest)
```

`bagboost`*bagboost*

Description

An implementation of the BagBoost algorithm for binary classification

Usage

```
bagboost(xlearn, ylearn, xtest, presel = 200, mfinal = 100, bag = 50)
```

Arguments

<code>xlearn</code>	A (n x p)-matrix, where rows correspond to training instances and columns contain the predictor variables.
<code>ylearn</code>	A vector of length n containing the class labels, which need to be coded by 0 and 1.
<code>xtest</code>	A (m x p)-matrix, where rows correspond to test instances and columns contain the predictor variables.
<code>presel</code>	An integer, giving the number of features to be pre-selected according to the Wilcoxon test statistic. Default is <code>presel=200</code> features. If <code>presel=0</code> , no feature preselection is carried out.
<code>mfinal</code>	An integer, the number of iterations for which boosting is run. Defaults to <code>mfinal=100</code> iterations
<code>bag</code>	An integer, the number of bagging steps that shall be done to obtain the weak learner. Defaults to <code>bag=50</code> bagging iterations.

Value

The function outputs an array, whose rows contain out-of-sample probabilities that the class labels are predicted as being of class 1, for every boosting iteration.

Author(s)

Marcel Dettling

References

- o "Boosting for Tumor Classification with Gene Expression Data", Marcel Dettling and Peter Bühlmann. *Bioinformatics* (2003), Vol. 19, p. 1061–1069.
- o "BagBoosting for Tumor Classification with Gene Expression Data", Marcel Dettling. To appear in *Bioinformatics* (2005).
- o Further information is available from the webpage <http://stat.ethz.ch/~dettling>

Examples

```

data(leukemia, package = "boost")

## 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]

## Classification with bagboost
fit <- bagboost(xlearn, ylearn, xtest, presel=50, mfinal=20, bag=5)
summarize(fit, ytest)

```

l2boost

l2boost

Description

An implementation of the LogitBoost algorithm for binary classification

Usage

```
l2boost(xlearn, ylearn, xtest, presel = 200, mfinal = 100)
```

Arguments

xlearn	A (n x p)-matrix, where rows correspond to training instances and columns contain the predictor variables.
ylearn	A vector of length n containing the class labels, which need to be coded by 0 and 1.
xtest	A (m x p)-matrix, where rows correspond to test instances and columns contain the predictor variables.
presel	An integer, giving the number of features to be pre-selected according to the Wilcoxon test statistic. Default is presel=200 features. If presel=0, no feature preselection is carried out.
mfinal	An integer, the number of iterations for which boosting is run. Defaults to mfinal=100 iterations

Value

The function outputs an array, whose rows contain out-of-sample probabilities that the class labels are predicted as being of class 1, for every boosting iteration.

Author(s)

Marcel Dettling

References

- o "Boosting for Tumor Classification with Gene Expression Data", Marcel Dettling and Peter Bühlmann. *Bioinformatics* (2003), Vol. 19, p. 1061–1069.
- o "BagBoosting for Tumor Classification with Gene Expression Data", Marcel Dettling. To appear in *Bioinformatics* (2005).
- o Further information is available from the webpage <http://stat.ethz.ch/~dettling>

Examples

```
data(leukemia, package = "boost")

## 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]

## Classification with l2boost
fit <- l2boost(xlearn, ylearn, xtest, presel=50, mfinal=20)
summarize(fit, ytest)
```

learner

Internal functions for the boost package.

Description

These are not to be called by the user.

See Also

simulator, logitboost, bagboost, adaboost, l2boost

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)
```

logitboost

logitboost

Description

An implementation of the LogitBoost algorithm for binary classification

Usage

```
logitboost(xlearn, ylearn, xtest, presel = 200, mfinal = 100)
```

Arguments

xlearn	A (n x p)-matrix, where rows correspond to training instances and columns contain the predictor variables.
ylearn	A vector of length n containing the class labels, which need to be coded by 0 and 1.
xtest	A (m x p)-matrix, where rows correspond to test instances and columns contain the predictor variables.
presel	An integer, giving the number of features to be pre-selected according to the Wilcoxon test statistic. Default is presel=200 features. If presel=0, no feature preselection is carried out.
mfinal	An integer, the number of iterations for which boosting is run. Defaults to mfinal=100 iterations

Value

The function outputs an array, whose rows contain out-of-sample probabilities that the class labels are predicted as being of class 1, for every boosting iteration.

Author(s)

Marcel Dettling

References

- o "Boosting for Tumor Classification with Gene Expression Data", Marcel Dettling and Peter Bühlmann. *Bioinformatics* (2003), Vol. 19, p. 1061–1069.
- o "BagBoosting for Tumor Classification with Gene Expression Data", Marcel Dettling. To appear in *Bioinformatics* (2005).
- o Further information is available from the webpage <http://stat.ethz.ch/~dettling>

Examples

```
data(leukemia, package = "boost")

## 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]

## Classification with logitboost
fit <- logitboost(xlearn, ylearn, xtest, presel=50, mfinal=20)
summarize(fit, ytest)
```

simulator

simulator

Description

Simulation of (microarray) data according to correlation and mean structures from real datasets.

Usage

```
simulator(x, y, respmod = c("none", "resp1", "resp2", "resp3"),
  nos = 1200, gene = NULL, signs = NULL)
```

Arguments

- | | |
|---------|--|
| x | A (n x p)-matrix, whose correlation and mean structure is to be used for simulating data. Its rows correspond to training instances and columns contain the predictor variables. |
| y | A vector of length n containing the class labels, which need to be coded by 0 and 1. |
| respmod | A character string. Either "none" where the simulated gene expression labels are determined model-free depending which class mean and correlation structure had been used for their determination. The choice of "resp1", "resp2" and "resp3" means that a response model is applied. For "resp1", 10 genes are selected and determine conditional probabilities via a logistic model with equal |

weights. The class labels are then regarded as having a Bernoulli distribution with probability p . For "resp2", 25 genes are plugged into the logistic model with non-equal weights. With "resp3", 25 genes are chosen for a logistic model with second and third order interactions.

nos	An integer, giving the number of instances which are simulated.
gene	A vector giving the index of the genes which shall be used for model based class label simulation. Defaults to NULL. This argument should only be used for specially designed simulation studies, where it is important that the same predictor variables are repeatedly used for simulating class label.
signs	A vector containing entries of +1 and -1. Defaults to NULL and is only of importance in specially designed simulation studies, where it is important that the same predictor variables are repeatedly used for simulating class label.

Details

The new instances are simulated according to a multivariate normal distribution with means and correlation structure taken from a real (gene expression) dataset. This structure is obtained by transforming a standard multivariate normal distribution, which requires an eigenvalue decomposition of the provided real dataset. For datasets with many predictors (>500), this can be fairly time consuming. Simulating data without applying a response model is fine for most purposes, only special analysis tasks usually require it.

Value

Returns a list containing

x	An (nos x p)-matrix, containing the simulated data
y	A vector of length nos, containing the class labels of the simulated data.
probab	A vector of length nos, containing the conditional probabilities of the simulated data. Is empty if respmod="none".
bayes	An integer, giving the Bayes error (theoretically minimal misclassification risk) for the simulated data. Is empty if respmod="none".
gene	A vector, containing the indices of the variables which had been used in the logistic model for either "resp1", "resp2" or "resp3". Is empty if respmod="none".
signs	A vector, containing -1 and +1. Indicates with what polarization a predictor variable had been used in the logistic model. Is empty if respmod="none".
b	

References

- o "BagBoosting for Tumor Classification with Gene Expression Data", Marcel Dettling. To appear in *Bioinformatics* (2005).
- o Further information is available from the webpage <http://stat.ethz.ch/~dettling>

Examples

```
set.seed(21)
data(leukemia)

## Simulation of gene expression data
simu <- simulator(leukemia.x, leukemia.y, nos=200)

## Defining training and test data
xlearn <- simu$x[1:150,]
ylearn <- simu$y[1:150]
xtest  <- simu$x[151:200,]
ytest  <- simu$y[151:200]

## Classification with logitboost
fit <- logitboost(xlearn, ylearn, xtest, mfinal=20, presel=50)
summarize(fit, ytest)
```

summarize

Summarize the output of classification with boosting functions

Description

Yields text and graphical output that summarizes the misclassification error rates that have been achieved with boosting methods

Usage

```
summarize(boost.out, resp, mout = ncol(boost.out), grafik = TRUE)
```

Arguments

<code>boost.out</code>	An R-object, as obtained from one of the functions 'bagboost', 'logitboost', 'adaboost' or 'l2boost'.
<code>resp</code>	A vector containing the class labels of the test instances. Needs to be coded by 0 and 1.
<code>mout</code>	The number of boosting iterations for which the error rate shall be printed. Defaults to the number of iterations boosting has been run for.
<code>grafik</code>	Logical, indicating whether a plot of the error rates is desired or not.

Value

Just verbatim and graphical output.

Author(s)

Marcel Dettling

References

- o "Boosting for Tumor Classification with Gene Expression Data", Marcel Dettling and Peter Bühlmann. *Bioinformatics* (2003), Vol. 19, p. 1061–1069.
- o "BagBoosting for Tumor Classification with Gene Expression Data", Marcel Dettling. To appear in *Bioinformatics* (2005).
- o Further information is available from the webpage <http://stat.ethz.ch/~dettling>

See Also

bagboost, logitboost, adaboost, l2boost

Examples

```
data(leukemia, package = "boost")

## 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]

## Classification with logitboost
fit <- logitboost(xlearn, ylearn, xtest, presel=50, mfinal=20)
summarize(fit, ytest)
```

Index

*Topic **classif**

- adaboost, 1
- bagboost, 2
- l2boost, 3
- learner, 5
- logitboost, 6
- simulator, 7
- summarize, 9

*Topic **datasets**

- leukemia, 5

adaboost, 1

bagboost, 2

l2boost, 3

learner, 5

leukemia, 5

logitboost, 6

response1 (*learner*), 5

response2 (*learner*), 5

response3 (*learner*), 5

score (*learner*), 5

simulator, 7

summarize, 9