Package ‘isoboost’

May 1, 2021

Title  Isotonic Boosting Classification Rules
Version  1.0.1
Date  2021-05-01
Description  In classification problems a monotone relation between some
             predictors and the classes may be assumed. In this package 'isoboost'
             we propose new boosting algorithms, based on LogitBoost, that
             incorporate this isotonicity information, yielding more accurate
             and easily interpretable rules.
Imports  Iso, isotone, rpart
License  GPL-2 | GPL-3
NeedsCompilation  no
Author  David Conde [aut, cre],
        Miguel A. Fernandez [aut],
        Cristina Rueda [aut],
        Bonifacio Salvador [aut]
Maintainer  David Conde <dconde@eio.uva.es>
Repository  CRAN
Date/Publication  2021-05-01 15:20:02 UTC

R topics documented:

isoboost-package ......................................................... 2
amilb ................................................................. 2
asilb ................................................................. 5
cmilb ................................................................. 7
csilb ................................................................. 9
motors ................................................................. 11

Index  14
isoboost-package  

*Isotonic Boosting Classification Rules*

**Description**

In this package we present new boosting classification rules based on LogitBoost when it can be assumed that higher (or lower) values of some predictors are related to higher levels of the response.

**Details**

- **Package**: isoboost
- **Type**: Package
- **Version**: 1.0.1
- **Date**: 2021-05-01
- **License**: GPL-2 | GPL-3

For a complete list of functions with individual help pages, use `library(help = "isoboost")`.

**Author(s)**

David Conde, Miguel A. Fernandez, Cristina Rueda, Bonifacio Salvador

Maintainer: David Conde <dconde@eio.uva.es>

---

amilb  

*(Adjacent-categories) Multiple Isotonic LogitBoost*

**Description**

Train and predict logitboost-based classification algorithm using multivariate isotonic regression (linear regression for no monotone features) as weak learners, based on the adjacent-categories logistic model (see Agresti (2010)). For full details on this algorithm, see Conde et al. (2020).

**Usage**

```r
amilb(xlearn, ...)  
amilb(formula, data, ...)  
amilb(xlearn, ylearn, xtest = xlearn, mfinal = 100, monotone_constraints = rep(0, dim(xlearn)[2]), prior = NULL, ...)
```

---

## S3 method for class 'formula'
amilb(formula, data, ...)

## Default S3 method:
amilb(xlearn, ylearn, xtest = xlearn, mfinal = 100, monotone_constraints = rep(0, dim(xlearn)[2]), prior = NULL, ...)
Arguments

- **formula**: A formula of the form $\text{groups} \sim x_1 + x_2 + \ldots$. That is, the response is the class variable and the right hand side specifies the explanatory variables.
- **data**: Data frame from which variables specified in formula are to be taken.
- **xlearn** (Required if no formula is given as the principal argument.) A data frame or matrix containing the explanatory variables.
- **ylearn** (Required if no formula is given as the principal argument.) A numeric vector or factor with numeric levels specifying the class for each observation.
- **xtest**: A data frame or matrix of cases to be classified, containing the features used in formula or xlearn.
- **mfinal**: Maximum number of iterations of the algorithm.
- **monotone_constraints**: Numerical vector consisting of 1, 0 and -1, its length equals the number of features in xlearn. 1 is increasing, -1 is decreasing and 0 is no constraint.
- **prior**: The prior probabilities of class membership. If unspecified, equal prior probabilities are used. If present, the probabilities must be specified in the order of the factor levels.
- **...**: Arguments passed to or from other methods.

Value

A list containing the following components:

- **call**: The (matched) function call.
- **trainset**: Matrix with the training set used (first columns) and the class for each observation (last column).
- **prior**: Prior probabilities of class membership used.
- **apparent**: Apparent error rate.
- **mfinal**: Number of iterations of the algorithm.
- **loglikelihood**: Log-likelihood.
- **posterior**: Posterior probabilities of class membership for xtest set.
- **class**: Labels of the class with maximal probability for xtest set.

Note

This function may be called using either a formula and data frame, or a data frame and grouping variable, or a matrix and grouping variable as the first two arguments. All other arguments are optional.

Classes must be identified, either in a column of data or in the ylearn vector, by natural numbers varying from 1 to the number of classes. The number of classes must be greater than 1.

If there are missing values in either data, xlearn or ylearn, corresponding observations will be deleted.
Author(s)

David Conde

References


See Also

asilb, csilb, cmilb

Examples

data(motors)
table(motors$condition)
## 1 2 3 4
## 83 67 70 60

## Let us consider the first three variables as predictors
data <- motors[, 1:3]
grouping = motors$condition
##
## Lower values of the amplitudes are expected to be
## related to higher levels of damage severity, so
## we can consider the following monotone constraints
monotone_constraints = rep(-1, 3)

set.seed(7964)
values <- runif(dim(data)[1])
trainsubset <- values < 0.2
obj <- amilb(data[trainsubset, ], grouping[trainsubset],
             data[-trainsubset, ], 100, monotone_constraints)

## Apparent error
obj$apparent
## 4.761905

## Error rate
100*mean(obj$class != grouping[-trainsubset])
## 15.41219
**asilb**

*(Adjacent-categories) Simple Isotonic LogitBoost*

**Description**

Train and predict logitboost-based classification algorithm using isotonic regression (decision stumps for no monotone features) as weak learners, based on the adjacent-categories logistic model (see Agresti (2010)). For full details on this algorithm, see Conde et al. (2020).

**Usage**

```r
asilb(xlearn, ...)  
## S3 method for class 'formula'  
asilb(formula, data, ...)  
## Default S3 method:  
asilb(xlearn, ylearn, xtest = xlearn, mfinal = 100,  
monotone_constraints = rep(0, dim(xlearn)[2]), prior = NULL, ...)  
```

**Arguments**

- `formula` A formula of the form `groups ~ x1 + x2 + ...`. That is, the response is the class variable and the right hand side specifies the explanatory variables.
- `data` Data frame from which variables specified in `formula` are to be taken.
- `xlearn` (Required if no formula is given as the principal argument.) A data frame or matrix containing the explanatory variables.
- `ylearn` (Required if no formula is given as the principal argument.) A numeric vector or factor with numeric levels specifying the class for each observation.
- `xtest` A data frame or matrix of cases to be classified, containing the features used in `formula` or `xlearn`.
- `mfinal` Number of iterations of the algorithm.
- `monotone_constraints` Numerical vector consisting of 1, 0 and -1, its length equals the number of features in `xlearn`. 1 is increasing, -1 is decreasing and 0 is no constraint.
- `prior` The prior probabilities of class membership. If unspecified, equal prior probabilities are used. If present, the probabilities must be specified in the order of the factor levels.
- `...` Arguments passed to or from other methods.

**Value**

A list containing the following components:

- `call` The (matched) function call.
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trainset</td>
<td>Matrix with the training set used (first columns) and the class for each observation (last column).</td>
</tr>
<tr>
<td>prior</td>
<td>Prior probabilities of class membership used.</td>
</tr>
<tr>
<td>apparent</td>
<td>Apparent error rate.</td>
</tr>
<tr>
<td>mfinal</td>
<td>Number of iterations of the algorithm.</td>
</tr>
<tr>
<td>loglikelihood</td>
<td>Log-likelihood.</td>
</tr>
<tr>
<td>posterior</td>
<td>Posterior probabilities of class membership for xtest set.</td>
</tr>
<tr>
<td>class</td>
<td>Labels of the class with maximal probability for xtest set.</td>
</tr>
</tbody>
</table>

**Note**

This function may be called using either a formula and data frame, or a data frame and grouping variable, or a matrix and grouping variable as the first two arguments. All other arguments are optional.

Classes must be identified, either in a column of data or in the ylearn vector, by natural numbers varying from 1 to the number of classes. The number of classes must be greater than 1.

If there are missing values in either data, xlearn or ylearn, corresponding observations will be deleted.

**Author(s)**

David Conde

**References**


**See Also**

amilb, csilb, cmilb

**Examples**

```r
data(motors)
table(motors$condition)
## 1 2 3 4
## 83 67 70 60

## Let us consider the first three variables as predictors
data <- motors[, 1:3]
grouping = motors$condition
##
## Lower values of the amplitudes are expected to be related to higher levels of damage severity, so we can consider the following monotone constraints```
monotone_constraints = rep(-1, 3)

set.seed(7964)
values <- runif(dim(data)[1])
trainsubset <- values < 0.2
obj <- asilb(data[trainsubset, ], grouping[trainsubset],
               data[-trainsubset, ], 50, monotone_constraints)

## Apparent error
obj$apparent
## 4.761905

## Error rate
100*mean(obj$class != grouping[-trainsubset])
## 14.69534

---

**cmilb**  
*Cumulative probabilities Multiple Isotonic LogitBoost*

### Description

Train and predict logitboost-based classification algorithm using multivariate isotonic regression (linear regression for no monotone features) as weak learners, based on the cumulative probabilities logistic model (see Agresti (2010)). For full details on this algorithm, see Conde et al. (2020).

### Usage

```r
cmilb(xlearn, ...)
```

**## S3 method for class 'formula'**

```r
cmilb(formula, data, ...)
```

**## Default S3 method:**

```r
cmilb(xlearn, ylearn, xtest = xlearn, mfinal = 100,
      monotone_constraints = rep(0, dim(xlearn)[2]), prior = NULL, ...)
```

### Arguments

- **formula**: A formula of the form `groups ~ x1 + x2 + ...`. That is, the response is the class variable and the right hand side specifies the explanatory variables.
- **data**: Data frame from which variables specified in `formula` are to be taken.
- **xlearn**: (Required if no formula is given as the principal argument.) A data frame or matrix containing the explanatory variables.
- **ylearn**: (Required if no formula is given as the principal argument.) A numeric vector or factor with numeric levels specifying the class for each observation.
- **xtest**: A data frame or matrix of cases to be classified, containing the features used in `formula` or `xlearn`. 
mfinal  Maximum number of iterations of the algorithm.
monotone_constraints  Numerical vector consisting of 1, 0 and -1, its length equals the number of features in xlearn. 1 is increasing, -1 is decreasing and 0 is no constraint.
prior  The prior probabilities of class membership. If unspecified, equal prior probabilities are used. If present, the probabilities must be specified in the order of the factor levels.

...  Arguments passed to or from other methods.

Value

A list containing the following components:
call  The (matched) function call.
trainset  Matrix with the training set used (first columns) and the class for each observation (last column).
prior  Prior probabilities of class membership used.
apparent  Apparent error rate.
mfinal  Number of iterations of the algorithm.
loglikelihood  Log-likelihood.
posterior  Posterior probabilities of class membership for xtest set.
class  Labels of the class with maximal probability for xtest set.

Note

This function may be called using either a formula and data frame, or a data frame and grouping variable, or a matrix and grouping variable as the first two arguments. All other arguments are optional.

Classes must be identified, either in a column of data or in the ylearn vector, by natural numbers varying from 1 to the number of classes. The number of classes must be greater than 1.

If there are missing values in either data, xlearn or ylearn, corresponding observations will be deleted.

Author(s)

David Conde

References


See Also

asilb, amilb, csilb
Examples

data(motors)
table(motors$condition)
## 1 2 3 4
## 83 67 70 60

## Let us consider the first three variables as predictors
data <- motors[, 1:3]
grouping = motors$condition

## Lower values of the amplitudes are expected to be
## related to higher levels of damage severity, so
## we can consider the following monotone constraints
monotone_constraints = rep(-1, 3)

set.seed(7964)
values <- runif(dim(data)[1])
trainsubset <- values < 0.2
obj <- cmilb(data[trainsubset, ], grouping[trainsubset],
             data[-trainsubset, ], 20, monotone_constraints)

## Apparent error
obj$apparent
## 4.761905

## Error rate
100*mean(obj$class != grouping[-trainsubset])
## 15.77061

---

**csilb**

*Cumulative probabilities Simple Isotonic LogitBoost*

**Description**

Train and predict logitboost-based classification algorithm using isotonic regression (decision stumps for no monotone features) as weak learners, based on the cumulative probabilities logistic model (see Agresti (2010)). For full details on this algorithm, see Conde et al. (2020).

**Usage**

csilb(xlearn, ...)

## S3 method for class 'formula'
csilb(formula, data, ...)

## Default S3 method:
csilb(xlearn, ylearn, xtest = xlearn, mfinal = 100,
monotone_constraints = rep(0, dim(xlearn)[2]), prior = NULL, ...)
Arguments

- **formula**: A formula of the form \( \text{groups} \sim x_1 + x_2 + \ldots \). That is, the response is the class variable and the right hand side specifies the explanatory variables.
- **data**: Data frame from which variables specified in `formula` are to be taken.
- **xlearn**: (Required if no formula is given as the principal argument.) A data frame or matrix containing the explanatory variables.
- **ylearn**: (Required if no formula is given as the principal argument.) A numeric vector or factor with numeric levels specifying the class for each observation.
- **xtest**: A data frame or matrix of cases to be classified, containing the features used in `formula` or `xlearn`.
- **mfinal**: Number of iterations of the algorithm.
- **monotone_constraints**: Numerical vector consisting of 1, 0 and -1, its length equals the number of features in `xlearn`. 1 is increasing, -1 is decreasing and 0 is no constraint.
- **prior**: The prior probabilities of class membership. If unspecified, equal prior probabilities are used. If present, the probabilities must be specified in the order of the factor levels.
- **...**: Arguments passed to or from other methods.

Value

A list containing the following components:

- **call**: The (matched) function call.
- **trainset**: Matrix with the training set used (first columns) and the class for each observation (last column).
- **prior**: Prior probabilities of class membership used.
- **apparent**: Apparent error rate.
- **mfinal**: Number of iterations of the algorithm.
- **loglikelihood**: Log-likelihood.
- **posterior**: Posterior probabilities of class membership for `xtest` set.
- **class**: Labels of the class with maximal probability for `xtest` set.

Note

This function may be called using either a formula and data frame, or a data frame and grouping variable, or a matrix and grouping variable as the first two arguments. All other arguments are optional.

Classes must be identified, either in a column of `data` or in the `ylearn` vector, by natural numbers varying from 1 to the number of classes. The number of classes must be greater than 1.

If there are missing values in either `data`, `xlearn` or `ylearn`, corresponding observations will be deleted.
Author(s)

David Conde

References


See Also

asilb, amilb, cmilb

Examples

data(motors)
table(motors$condition)
## 1 2 3 4
## 83 67 70 60

## Let us consider the first three variables as predictors
data <- motors[, 1:3]
grouping = motors$condition
##
## Lower values of the amplitudes are expected to be related to higher levels of damage severity, so we can consider the following monotone constraints
monotone_constraints = rep(-1, 3)

set.seed(7964)
values <- runif(dim(data)[1])
trainsubset <- values < 0.2
obj <- csilb(data[trainsubset, ], grouping[trainsubset],
            data[-trainsubset, ], 100, monotone_constraints)

## Apparent error
obj$apparent
## 4.761905

## Error rate
100*mean(obj$class != grouping[-trainsubset])
## 17.92115

motors

*Diagnostic of electrical induction motors*
Description

Electrical induction motors are widely used in industry. In the industrial context, the early detection of possible damage in the motor is very important since failures can result in financial losses. Motor Current Signature Analysis is the most widespread technique to diagnose a faulty motor, see Choudhary et al. (2019). This technique is based on the spectral analysis of the stator current: motor faults cause an asymmetry that reflects as additional harmonics in the current spectrum, so side bands around the main frequency are considered and amplitudes of these side bands around odd harmonics are measured.

The data were generated by Oscar Duque and Daniel Morinigo at the Electrical Engineering laboratory of the Universidad de Valladolid.

Four condition states of damage severity are considered: 1 - undamaged, 2 - incipient fault, 3 - moderate damage, 4 - severe damage.

Usage

data(motors)

Format

A data frame with 280 observations on 7 variables, six are numerical and one nominal defining the condition state of the motors.

<table>
<thead>
<tr>
<th></th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>amplitude_l.1</td>
<td>Amplitude of the first lower side band around harmonic 1</td>
</tr>
<tr>
<td>[2]</td>
<td>amplitude_u.1</td>
<td>Amplitude of the first upper side band around harmonic 1</td>
</tr>
<tr>
<td>[3]</td>
<td>amplitude_l.5</td>
<td>Amplitude of the first lower side band around harmonic 5</td>
</tr>
<tr>
<td>[4]</td>
<td>amplitude_u.5</td>
<td>Amplitude of the first upper side band around harmonic 5</td>
</tr>
<tr>
<td>[5]</td>
<td>amplitude_l.7</td>
<td>Amplitude of the first lower side band around harmonic 7</td>
</tr>
<tr>
<td>[6]</td>
<td>amplitude_u.7</td>
<td>Amplitude of the first upper side band around harmonic 7</td>
</tr>
<tr>
<td>[7]</td>
<td>condition</td>
<td>Condition state</td>
</tr>
</tbody>
</table>

Source

- Creator: Oscar Duque and Daniel Morinigo, Electrical Engineering Department laboratory, Universidad de Valladolid, Valladolid, Spain.

References


Examples

data(motors)
motors

summary(motors)
Index

* classif
  amilb, 2
  asilb, 5
  cmilb, 7
  csilb, 9
  isoboost-package, 2

* datasets
  motors, 11

* multivariate
  amilb, 2
  asilb, 5
  cmilb, 7
  csilb, 9
  isoboost-package, 2

* package
  isoboost-package, 2

* tree
  isoboost-package, 2

amilb, 2, 6, 8, 11
asilb, 4, 5, 8, 11

cmilb, 4, 6, 7, 11
csilb, 4, 6, 8, 9

isoboost/isoboost-package, 2
isoboost-package, 2

motors, 11