Mars

Multivariate Adaptive Regression Splines

Description

Multivariate Adaptive Regression Splines from Friedman's "Multivariate Adaptive Regression Splines" (1991). Builds linear regression models at hinges.

Usage

```
mars(formula, data, control)
```

Arguments

Argument	Description
`formula`	Formula for multivariate adaptive regression
`data`	Data for running MARS on
`control`	Helper function that calls on constructor and validator

Details

Implementation of the algorithms and techniques from Friedman's "Multivariate Adaptive Regression Splines" (1991).

MARS is an extension of lm(), taking a symbolically specified formula with a response vector 'y' and input matrix 'x' and returning model parameters that account for nonlinearity and interaction between variables.

MARS uses a modified version of the forward stepwise algorithm used in traditional recursive partitioning. A truncated power spline function replaces the step function from recursive partitioning and the parent basis function remains eligible for further splitting alongside its children. MARS' forward stepwise algorithm restricts basis function products to factors involving distinct predictor variables and produces product spline basis functions with knots at all marginal data values.

The subset of basis functions produced from the forward stepwise algorithm is then subjected to a one-at-a-time backward stepwise function which creates a series of models with each new model having one less basis function than the last. The model with the best fit is returned.

MARS implements components of spline fitting and recursive partitioning to provide a flexible regression modeling technique for high dimensional data.

Value

`mars` returns an object of class `"mars"` as a list containing these unique values + values from lm objects.

The functions `summary` and `anova` are used to obtain and print a summary and analysis of variance table of the results. The generic accessor functions `plot`, `predict`, `print`, `fitted`, and `residuals` extract various

useful features of the value returned by `mars`.

All `"mars"` object inherit `"lm"` objects, thus can use `lm` methods.

An object of class `"mars"` is a list containing at least the following components:

`y`	Response variable used in the MARS formula.
`B`	Basis functions that survive after forward and backwards stepwise.
`splits`	Splits of each iteration are recorded from the forward and backward stepwise.
`formula`	The regression formula called.
`data`	The dataset used in the formula.
`mc`	Helper function values.
`mf`	The predictors of the model.
`coefficients`	A named vector of coefficients.
`residuals`	The residuals, that is response minus fitted values.
`effects`	Rotated response values according to the QR factorization for design matrix
`rank`	The numeric rank of the fitted model.
`fitted.values`	The fitted mean values.
`df.residual`	The residual degrees of freedom.
`xlevels`	A record of the levels of the factors used in fitting.
`call`	The matched call.
`terms`	The `terms` object used.
`model`	The model dataframe.

References

Friedman's "Multivariate Adaptive Regression Splines" (1991)

R documentation for lm function

R Documentation for Package 'earth'

Documenting functions manual

See also

`anova.mars` Anova decomposition method for objects of class mars. Outputs the variances of the basis functions.

`print.mars` Print method for objects of class mars. Outputs the call and Rss, GCV, GRsq, and Rsq.

`summary.mars` Summary method for objects of class mars. Works like print except also outputs the coefficients at the respective hinges.

`predict.mars` Predict method for objects of class mars. If no new data is provided, it will return the fitted values. If data is provided, it outputs the fitted values with the new data input.

`plot.mars` Plots method for objects of class mars. Outputs residuals vs fitted, response vs explanatory with fitted points, qq plot and

`residuals`, `fitted` Residuals method for objects of class mars implemented from lm. Outputs the residuals from the model. Fitted method for objects of class mars implemented from lm. Outputs the fitted values from the model.

See `lm documentation` for more details as mars inherit from lm objects.

See `Benson's Github Page` to download the source files and run the examples.

Examples

```
# source files from https://github.com/bensonouyang/MARS
source(mars.R)
source(anova.R)
source(plot.R)
source(predict.R)
source(print.R)
source(summary.R)

## Example 1

library(ISLR)
data(Wage)
mc <- mars.control(Mmax=10)
mout <- mars(wage ~ age + education, data=Wage, control=mc)
ff <- fitted(mout)
p1 <- predict(mout)
p2 <- predict(mout,newdata=data.frame(age=Wage$age,education=Wage$education))
head(cbind(ff,p1,p2)) # columns should be identical
mout # tests print method
summary(mout) #test summary method
anova(mout) # test anova method
plot(mout) # test plot method


## Example 2

### data gathered from
### https://www.kaggle.com/ucsandiego/carbon-dioxide
### imported cleaned data

archive = read.csv("archive.csv")
train_data = data.frame(y = archive$Year, x = archive$Carbon.Dioxide..ppm.)
mc <- mars.control(Mmax=2)
mout <- mars(y ~ x, data=train_data, control=mc)
ff <- fitted(mout)
p1 <- predict(mout)
p2 <- predict(mout,newdata=data.frame(x=train_data$x))
head(cbind(ff,p1,p2)) # columns should be identical
mout # tests print method
summary(mout) #test summary method
anova(mout) # test anova method
```

```
plot(mout) # test plot method


## Example 3

### data gathered from
# https://www.kaggle.com/dgrechka/covid19-transmission-periods-per-week-per-country?select=params.csv
### imported cleaned data

params = read.csv("params.csv")
mc <- mars.control(Mmax=2)
mout <- mars(PeakDayNum ~ FirstDayNum, data=params, control=mc)
ff <- fitted(mout)
p1 <- predict(mout)
p2 <- predict(mout,newdata=data.frame(x=train_data2$x))
head(cbind(ff,p1,p2)) # columns should be identical
mout # tests print method
summary(mout) #test summary method
anova(mout) # test anova method
plot(mout) # test plot method
```