Package ‘mars’

April 12, 2022

Title Multiple Adaptive Regression Spline

Version 0.0.0.9000

Description Implimentation of Friedman's Multiple Adaptive Regression Spline (MARS) model.

Maintainer Gareth Bennett <garethb@sfu.ca>

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Imports stats

Suggests knitr,
rmarkdown,
testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 2.10)

LazyData true

R topics documented:

mars *Multivariate Adaptive Regression Splines (MARS)*

Description

Fit Friedman's Multivariate Adaptive Regression Splines (MARS) model.

Usage

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

Arguments

formula	an R formula
data	a data frame containing your data
control	an optional object of class 'mars.control'

Details

The MARS function attempts to fit non-linear data as accurately as possible. To do this the function defines one or more hinge-points that act as connectors between two or more linear functions.This process is completed through a multistage process including a forward pass and backwards pass.

The forward pass algorithm takes a formula and data as input and attempts to output a matrix of basis functions of size Mmax. To do this, the algorithm considers each data point for each predictor and generates candidate basis function pairs. These candidates are selected based on reduction to the model's residual sum-of-squares. The basis function matrix contains a row for each observations and a column for each basis function.

The forward pass identifies every basis function that reduces the model's residual sum-of-squares, but this can produce an over-fit model. To create a model that generalized better with new data, the matrix is handed to the backwards pass algorithm. The algorithm 'prunes' any basis function that adds no significant predictive accuracy to the overall model. The backwards algorithm compares subsets of basis function using generalized cross-validation scores in which 'd' controls the amount of penalization there is for extra model terms. By tracking GCV scores for each possible combination of basis function, the best possible model can be selected. The backwards pass algorithm updates the basis matrix to contain only the function from the optimal model.

Value

an object of class 'mars' which can be passed to plot,predict,summary and print methods.

Author(s)

Gareth Bennett, Heewon Oh, Jusung Lee

References

Jerome H. Friedman. Multivariate Adaptive Regression Splines (with discussion).Annals of Statistics 19/1, 1991. https://statistics.stanford.edu/research/multivariate-adaptive-regression-splines.

See Also

mars.control for constructing control objects

plot.mars for plotting results

predict.mars for predictions

summary.mars for summarizing mars objects

print.mars for printing mars objects

Examples

```
mm<-mars(y~.,dat=mars::marstestdata)
```

`mars.control` *Mars Control Object*

Description

Constructor for `mars.control` objects

This function constructs a `mars.control` object that specifies parameters used in the model fitting procedure

Usage

```
mars.control(Mmax = 2, d = 3, trace = FALSE)
```

Arguments

`Mmax`	Maximum number of basis functions. Should be an even interger Default value is 1.
`d`	The coefficient in the penalty term of the generalized cross validation measure. Default is 3.
`trace`	Should we print information about the fitting? Default is `FALSE`

Value

a `mars.control` object

Examples

```
mc<-mars.control(Mmax=10)
```

marstestdata *A test dataset for the mars package.*

Description

A dataset of size N=100 with n=10 explanatory variables, and a response variable that depends on only the first two explanatory variables.

Usage

```
marstestdata
```

Format

A data frame with 100 rows and 11 variables:

y response variable

x1 explanatory variable

x2 explanatory variable

x3 explanatory variable

x4 explanatory variable

x5 explanatory variable

x6 explanatory variable

x7 explanatory variable

x8 explanatory variable

x9 explanatory variable

x10 explanatory variable

plot.mars *Plot a mars object*

Description

Plots the fitted basis functions that depend on explanatory variable(main effects) or two explanatory variables (two-way interactions). Use `predict.lm()` to see the residual plots

Usage

```
## S3 method for class 'mars'
plot(x, ...)
```

Arguments

x	a mars x
...	additional arguments to pass to plot()

See Also

Other methods: predict.mars(), print.mars(), summary.mars()

Examples

```
mm<-mars(y~x1+x2,data=marstestdata,mars.control(Mmax=4))
plot(mm,col="red")
class(mm)<-"lm";plot(mm) #Standard diagnostic plots
```

---

predict.mars *Predict Function*

Description

Predict Function

Usage

```
## S3 method for class 'mars'
predict(object, newdata, ...)
```

Arguments

object	a mars object
newdata	optional new data for which predictions are required
...	additional arguments to predict()–currently not used

Value

the corresponding matrix of basis functions

See Also

Other methods: plot.mars(), print.mars(), summary.mars()

Examples

```
mm<- mars(y~x1+x2,dat=mars::marstestdata)
predict(mm,newdata=data.frame(x1=rnorm(100),x2=rnorm(100)))
```

print.mars *Print a Mars Object*

Description

Print a Mars Object

Usage

```
## S3 method for class 'mars'
print(x, ...)
```

Arguments

x	a mars object
...	additional arguments to print.mars, currently unused

See Also

Other methods: plot.mars(), predict.mars(), summary.mars()

Examples

```
mm <- mars(y~x1+x2,data=marstestdata)
print(mm)
```

summary.mars *Summary of Mars Object*

Description

summary method for class "mars". Prints a summary of hinges that make up each basis function in the optimal model, along with the coefficient of said basis functions

Usage

```
## S3 method for class 'mars'
summary(object, ...)
```

Arguments

object	an object of class "mars", usually a result of a call to mars
...	additional arguments to print.mars, currently unused

Value

a summary of the mars object

Author(s)

Gareth Bennett, Heewon Oh, Jusung Lee

See Also

Other methods: `plot.mars()`, `predict.mars()`, `print.mars()`

Examples

```
mm<-mars(y ~.,data=mars::marstestdata)
summary(mm)
```

Index