

Package ‘acepack’

April 6, 2012

Maintainer Jonathan Baron <baron@psych.upenn.edu>

Version 1.3-3.1

Author Phil Spector, Jerome Friedman, Robert Tibshirani, Thomas Lumley

Description ACE and AVAS methods for choosing regression transformations.

Title ace() and avas() for selecting regression transformations

License MIT + file LICENSE

Repository CRAN

Date/Publication 2012-04-06 05:45:08

R topics documented:

ace	1
avas	3

Index	6
--------------	----------

ace	<i>Alternating Conditional Expectations</i>
-----	---

Description

Uses the alternating conditional expectations algorithm to find the transformations of y and x that maximise the proportion of variation in y explained by x.

Usage

```
ace(x, y, wt = rep(1, nrow(x)), cat = NULL, mon = NULL, lin = NULL,  
    circ = NULL, delrsq = 0.01)
```

Arguments

<code>x</code>	a matrix containing the independent variables.
<code>y</code>	a vector containing the response variable.
<code>wt</code>	an optional vector of weights.
<code>cat</code>	an optional integer vector specifying which variables assume categorical values. Positive values in <code>cat</code> refer to columns of the <code>x</code> matrix and zero to the response variable. Variables must be numeric, so a character variable should first be transformed with <code>as.numeric()</code> and then specified as categorical.
<code>mon</code>	an optional integer vector specifying which variables are to be transformed by monotone transformations. Positive values in <code>mon</code> refer to columns of the <code>x</code> matrix and zero to the response variable.
<code>lin</code>	an optional integer vector specifying which variables are to be transformed by linear transformations. Positive values in <code>lin</code> refer to columns of the <code>x</code> matrix and zero to the response variable.
<code>circ</code>	an integer vector specifying which variables assume circular (periodic) values. Positive values in <code>circ</code> refer to columns of the <code>x</code> matrix and zero to the response variable.
<code>delrsq</code>	termination threshold. Iteration stops when R-squared changes by less than <code>delrsq</code> in 3 consecutive iterations (default 0.01).

Value

A structure with the following components:

<code>x</code>	the input <code>x</code> matrix.
<code>y</code>	the input <code>y</code> vector.
<code>tx</code>	the transformed <code>x</code> values.
<code>ty</code>	the transformed <code>y</code> values.
<code>rsq</code>	the multiple R-squared value for the transformed values.
<code>l</code>	the codes for <code>cat</code> , <code>mon</code> , ...
<code>m</code>	not used in this version of <code>ace</code>

References

Breiman and Friedman, *Journal of the American Statistical Association* (September, 1985).

The R code is adapted from S code for `avas()` by Tibshirani, in the Statlib S archive; the FORTRAN is a double-precision version of FORTRAN code by Friedman and Spector in the Statlib general archive.

Examples

```
TWOPI <- 8*atan(1)
x <- runif(200,0,TWOPI)
y <- exp(sin(x)+rnorm(200)/2)
a <- ace(x,y)
```

```

par(mfrow=c(3,1))
plot(a$y,a$ty) # view the response transformation
plot(a$x,a$tx) # view the carrier transformation
plot(a$tx,a$ty) # examine the linearity of the fitted model

# From D. Wang and M. Murphy (2005), Identifying nonlinear relationships
# regression using the ACE algorithm. Journal of Applied Statistics,
# 32, 243-258.
X1 <- runif(100)*2-1
X2 <- runif(100)*2-1
X3 <- runif(100)*2-1
X4 <- runif(100)*2-1

# Original equation of Y:
Y <- log(4 + sin(3*X1) + abs(X2) + X3^2 + X4 + .1*rnorm(100))

# Transformed version so that Y, after transformation, is a
# linear function of transforms of the X variables:
# exp(Y) = 4 + sin(3*X1) + abs(X2) + X3^2 + X4

a1 <- ace(cbind(X1,X2,X3,X4),Y)

par(mfrow=c(2,1))

# For each variable, show its transform as a function of
# the original variable and the of the transform that created it,
# showing that the transform is recovered.
plot(X1,a1$tx[,1])
plot(sin(3*X1),a1$tx[,1])

plot(X2,a1$tx[,2])
plot(abs(X2),a1$tx[,2])

plot(X3,a1$tx[,3])
plot(X3^2,a1$tx[,3])

plot(X4,a1$tx[,4])
plot(X4,a1$tx[,4])

plot(Y,a1$ty)
plot(exp(Y),a1$ty)

```

Description

Estimate transformations of x and y such that the regression of y on x is approximately linear with constant variance

Usage

```
avas(x, y, wt = rep(1, nrow(x)), cat = NULL, mon = NULL,
     lin = NULL, circ = NULL, delrsq = 0.01, yspan = 0)
```

Arguments

<code>x</code>	a matrix containing the independent variables.
<code>y</code>	a vector containing the response variable.
<code>wt</code>	an optional vector of weights.
<code>cat</code>	an optional integer vector specifying which variables assume categorical values. Positive values in <code>cat</code> refer to columns of the <code>x</code> matrix and zero to the response variable. Variables must be numeric, so a character variable should first be transformed with <code>as.numeric()</code> and then specified as categorical.
<code>mon</code>	an optional integer vector specifying which variables are to be transformed by monotone transformations. Positive values in <code>mon</code> refer to columns of the <code>x</code> matrix and zero to the response variable.
<code>lin</code>	an optional integer vector specifying which variables are to be transformed by linear transformations. Positive values in <code>lin</code> refer to columns of the <code>x</code> matrix and zero to the response variable.
<code>circ</code>	an integer vector specifying which variables assume circular (periodic) values. Positive values in <code>circ</code> refer to columns of the <code>x</code> matrix and zero to the response variable.
<code>delrsq</code>	termination threshold. Iteration stops when R-squared changes by less than <code>delrsq</code> in 3 consecutive iterations (default 0.01).
<code>yspan</code>	Optional window size parameter for smoothing the variance. Range is $[0, 1]$. Default is 0 (cross validated choice). <code>.5</code> is a reasonable alternative to try.

Value

A structure with the following components:

<code>x</code>	the input <code>x</code> matrix.
<code>y</code>	the input <code>y</code> vector.
<code>tx</code>	the transformed <code>x</code> values.
<code>ty</code>	the transformed <code>y</code> values.
<code>rsq</code>	the multiple R-squared value for the transformed values.
<code>l</code>	the codes for <code>cat</code> , <code>mon</code> , ...
<code>m</code>	not used in this version of <code>avas</code>
<code>yspan</code>	span used for smoothing the variance
<code>iters</code>	iteration number and <code>rsq</code> for that iteration
<code>niters</code>	number of iterations used

References

Rob Tibshirani (1987), “Estimating optimal transformations for regression”. *Journal of the American Statistical Association* **83**, 394ff.

Examples

```
TWOPI <- 8*atan(1)
x <- runif(200,0,TWOPI)
y <- exp(sin(x)+rnorm(200)/2)
a <- avas(x,y)
par(mfrow=c(3,1))
plot(a$y,a$ty) # view the response transformation
plot(a$x,a$tx) # view the carrier transformation
plot(a$tx,a$ty) # examine the linearity of the fitted model

# From D. Wang and M. Murphy (2005), Identifying nonlinear relationships
# regression using the ACE algorithm. Journal of Applied Statistics,
# 32, 243-258, adapted for avas.
X1 <- runif(100)*2-1
X2 <- runif(100)*2-1
X3 <- runif(100)*2-1
X4 <- runif(100)*2-1

# Original equation of Y:
Y <- log(4 + sin(3*X1) + abs(X2) + X3^2 + X4 + .1*rnorm(100))

# Transformed version so that Y, after transformation, is a
# linear function of transforms of the X variables:
# exp(Y) = 4 + sin(3*X1) + abs(X2) + X3^2 + X4

a1 <- avas(cbind(X1,X2,X3,X4),Y)

par(mfrow=c(2,1))

# For each variable, show its transform as a function of
# the original variable and the of the transform that created it,
# showing that the transform is recovered.
plot(X1,a1$tx[,1])
plot(sin(3*X1),a1$tx[,1])

plot(X2,a1$tx[,2])
plot(abs(X2),a1$tx[,2])

plot(X3,a1$tx[,3])
plot(X3^2,a1$tx[,3])

plot(X4,a1$tx[,4])
plot(X4,a1$tx[,4])

plot(Y,a1$ty)
plot(exp(Y),a1$ty)
```

Index

*Topic **models**

ace, [1](#)

avas, [3](#)

ace, [1](#)

avas, [3](#)