Package 'acepack'

October 12, 2022

Maintainer Shawn Garbett <shawn.garbett@vanderbilt.edu>

Version 1.4.1

Author Phil Spector, Jerome Friedman, Robert Tibshirani, Thomas Lumley, Shawn Garbett, Jonathan Baron

Description Two nonparametric methods for multiple regression transform selection are provided. The first, Alternative Conditional Expectations (ACE), is an algorithm to find the fixed point of maximal correlation, i.e. it finds a set of transformed response variables that maximizes R^2 using smoothing functions [see Breiman, L., and J.H. Friedman. 1985. ``Estimating Optimal Transformations for Multiple Regression and Correlation''. Journal of the American Statistical Association. 80:580-598. <doi:10.1080/01621459.1985.10478157>]. Also included is the Additivity Variance Stabilization (AVAS) method which works better than ACE when correlation is low [see Tibshirani, R.. 1986. ``Estimating Transformations for Regression via Additivity and Variance Stabilization''. Journal of the American Statistical Association. 83:394-405. <doi:10.1080/01621459.1988.10478610>]. A good introduction to these two methods is in chapter 16 of Frank Harrel's ``Regression Modeling Strategies'' in the Springer Series in Statistics.

Title ACE and AVAS for Selecting Multiple Regression Transformations

License MIT + file LICENSE

Suggests testthat

Repository CRAN

Date/Publication 2016-10-29 00:11:52

NeedsCompilation yes

R topics documented:

ace *Alternating Conditional Expectations*

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. When x is a matrix, it is transformed so that its columns are equally weighted when predicting y.

Usage

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

Arguments

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

Value

A structure with the following components:

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

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

# example when x is a matrix
X1 <- 1:10
X2 <- X1^2
X <- cbind(X1,X2)
Y <- 3*X1+X2
a1 <- ace(X,Y)
plot(rowSums(a1$tx),a1$y)
(lm(a1$y ~ a1$tx)) # shows that the colums of X are equally weighted

# 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)

# 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.
par(mfrow=c(2,1))

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)
```

---

avas *Additivity and variance stabilization for regression*

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

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

Value

A structure with the following components:

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