Package ‘leaps’

May 5, 2009

Title regression subset selection

Version 2.9

Author Thomas Lumley <tlumley@u.washington.edu> using Fortran code by Alan Miller

Description Regression subset selection including exhaustive search

Depends

Suggests biglm

License GPL (>= 2)

Maintainer Thomas Lumley <tlumley@u.washington.edu>

Repository CRAN

Date/Publication 2009-05-05 18:43:21

R topics documented:

leaps *all-subsets regressiom*

Description

leaps() performs an exhaustive search for the best subsets of the variables in x for predicting y in linear regression, using an efficient branch-and-bound algorithm. It is a compatibility wrapper for regsubsets does the same thing better.

Since the algorithm returns a best model of each size, the results do not depend on a penalty model for model size: it doesn't make any difference whether you want to use AIC, BIC, CIC, DIC, ...

Usage

```
leaps(x=, y=, wt=rep(1, NROW(x)), int=TRUE, method=c("Cp", "adjr2", "r2"), nbest=10
```

Arguments

x A matrix of predictors

y A response vector

wt Optional weight vector

int Add an intercept to the model

method Calculate Cp, adjusted R-squared or R-squared

nbest Number of subsets of each size to report

names vector of names for columns of x

df Total degrees of freedom to use instead of nrow(x) in calculating Cp and adjusted R-squared

strictly.compatible
Implement misfeatures of leaps() in S

Value

A list with components

which logical matrix. Each row can be used to select the columns of x in the respective model

size Number of variables, including intercept if any, in the model

cp or adjr2 or r2 is the value of the chosen model selection statistic for each model

label vector of names for the columns of x

Note

With strictly.compatible=T the function will stop with an error if x is not of full rank or if it has more than 31 columns. It will ignore the column names of x even if names==NULL and will replace them with "0" to "9", "A" to "Z".

References

Alan Miller "Subset Selection in Regression" Chapman & Hall

See Also

`regsubsets`, `regsubsets.formula`, `regsubsets.default`

Examples

```
x<-matrix(rnorm(100),ncol=4)
y<-rnorm(25)
leaps(x,y)
```

`leaps.setup` *Internal functions for leaps(), subsets()*

Description

These functions are used internally by `regsubsets` and `leaps`. They are wrappers for Fortran routines that construct and manipulate a QR decomposition.

Usage

```
leaps.setup(x,y,wt=rep(1,length(y)),force.in=NULL,force.out=NULL,intercept=TRUE,nvm
leaps.seqrep(leaps.obj)
leaps.exhaustive(leaps.obj,really.big=FALSE)
leaps.backward(leaps.obj)
leaps.forward(leaps.obj)
```

Arguments

`x`	A matrix of predictors
`y`	A response vector
`wt`	Optional weight vector
`intercept`	Add an intercept to the model
`force.in`	vector indicating variable that must be in the model
`force.out`	vector indicating variable that must not be in the model
`nbest`	Number of subsets of each size to report
`nvmax`	largest subset size to examine
`warn.dep`	warn if `x` is not of full rank
`leaps.obj`	An object of class `leaps` as produced by `leaps.setup`
`really.big`	required before R gets sent off on a long uninterruptible computation

See Also

`regsubsets`, `leaps`

`plot.regsubsets` *Graphical table of best subsets*

Description

Plots a table of models showing which variables are in each model. The models are ordered by the specified model selection statistic. This plot is particularly useful when there are more than ten or so models and the simple table produced by `summary.regsubsets` is too big to read.

Usage

```
## S3 method for class 'regsubsets':
plot(x, labels=obj$xnames, main=NULL, scale=c("bic", "Cp", "adjr2", "r2"), col=gray
```

Arguments

`x`	`regsubsets` object
`labels`	variable names
`main`	title for plot
`scale`	which summary statistic to use for ordering plots
`col`	Colors: the last color should be close to but distinct from white
`...`	other arguments

Value

None

Author(s)

Thomas Lumley, based on a concept by Merlise Clyde

See Also

`regsubsets`,`summary.regsubsets`

Examples

```
data(swiss)
a<-regsubsets(Fertility~.,nbest=3,data=swiss)
par(mfrow=c(1,2))
plot(a)
plot(a,scale="r2")
```

regsubsets *functions for model selection*

Description

Model selection by exhaustive search, forward or backward stepwise, or sequential replacement

Usage

```
regsubsets(x=, ...)

## S3 method for class 'formula':
regsubsets(x=, data=, weights=NULL, nbest=1, nvmax=8, force.in=NULL, force.out=NULL

## Default S3 method:
regsubsets(x=, y=, weights=rep(1, length(y)), nbest=1, nvmax=8,
force.in=NULL, force.out=NULL, intercept=TRUE, method=c("exhaustive",
"backward", "forward", "seqrep"), really.big=FALSE,...)

## S3 method for class 'biglm':
regsubsets(x,nbest=1,nvmax=8,force.in=NULL,
method=c("exhaustive","backward", "forward", "seqrep"), really.big=FALSE,...)

## S3 method for class 'regsubsets':
summary(object,all.best=TRUE,matrix=TRUE,matrix.logical=FALSE,df=NULL,...)

## S3 method for class 'regsubsets':
coef(object,id,vcov=FALSE,...)
## S3 method for class 'regsubsets':
vcov(object,id,...)
```

Arguments

`x`	design matrix or model formula for full model, or `biglm` object
`data`	Optional data frame
`y`	response vector
`weights`	weight vector
`nbest`	number of subsets of each size to record
`nvmax`	maximum size of subsets to examine
`force.in`	index to columns of design matrix that should be in all models
`force.out`	index to columns of design matrix that should be in no models
`intercept`	Add an intercept?
`method`	Use exhaustive search, forward selection, backward selection or sequential replacement to search.

`really.big` Must be TRUE to perform exhaustive search on more than 50 variables.

`object` regsubsets object

`all.best` Show all the best subsets or just one of each size

`matrix` Show a matrix of the variables in each model or just summary statistics

`matrix.logical`
With `matrix=TRUE`, the matrix is logical `TRUE`/`FALSE` or string `"*"/" "`

`df` Specify a number of degrees of freedom for the summary statistics. The default is `n-1`

`id` Which model or models (ordered as in the summary output) to return coefficients and variance matrix for

`vcov` If `TRUE`, return the variance-covariance matrix as an attribute

`...` Other arguments for future methods

Details

Since this function returns separate best models of all sizes up to `nvmax` and since different model selection criteria such as AIC, BIC, CIC, DIC, ... differ only in how models of different sizes are compared, the results do not depend on the choice of cost-complexity tradeoff.

When `x` is a `biglm` object it is assumed to be the full model, so `force.out` is not relevant. If there is an intercept it is forced in by default; specify a `force.in` as a logical vector with `FALSE` as the first element to allow the intercept to be dropped.

The model search does not actually fit each model, so the returned object does not contain coefficients or standard errors. Coefficients and the variance-covariance matrix for one or model models can be obtained with the `coef` and `vcov` methods.

Value

`regsubsets` returns an object of class "regsubsets" containing no user-serviceable parts. It is designed to be processed by `summary.regsubsets`.

`summary.regsubsets` returns an object with elements

`which` A logical matrix indicating which elements are in each model

`rsq` The r-squared for each model

`rss` Residual sum of squares for each model

`adjr2` Adjusted r-squared

`cp` Mallows' Cp

`bic` Schwartz's information criterion, BIC

`outmat` A version of the `which` component that is formatted for printing

`obj` A copy of the `regsubsets` object

The `coef` method returns a coefficient vector or list of vectors, the `vcov` method returns a matrix or list of matrices.

See Also

`leaps`

Examples

```
data(swiss)
a<-regsubsets(as.matrix(swiss[,-1]),swiss[,1])
summary(a)
b<-regsubsets(Fertility~.,data=swiss,nbest=2)
summary(b)

coef(a, 1:3)
vcov(a, 3)
```

Index