Package 'biglars'

February 19, 2015

Type Package

Title Scalable Least-Angle Regression and Lasso

Version 1.0.2

Date Tue Dec 27 15:06:08 PST 2011

Author Mark Seligman, Chris Fraley, Tim Hesterberg

Maintainer Mark Seligman <Mark.Seligman@insilicos.com>

Description Least-angle regression, lasso and stepwise regression for numeric datasets in which the number of observations is greater than the number of predictors. The functions can be used with the ff library to accomodate datasets that are too large to be held in memory.

Depends R (>= 2.10), ff

License GPL (>= 2)

Repository CRAN

Date/Publication 2011-12-28 13:50:26

NeedsCompilation yes

R topics documented:

biglars.fit 2
diabetes 3
qrBlockApply 5

Index **6**

`biglars.fit` *Least-Angle, Lasso and Stepwise Regression*

Description

Least-angle, lasso and stepwise methods for linear regression.

Usage

```
biglars.fit(x, y, type = "lasso", removeColumns = TRUE,
    eps = sqrt(.Machine$double.eps), blockSize = NULL, maxStages = NULL)
```

Arguments

`x`	A matrix or `ff` of numeric predictors. The number of columns should not be larger than the number of rows.
`y`	A numeric response vector or 1-column `ff`.
`type`	The type of regression to be performed. The usual choices are **lasso** lasso method (L1 penalty) **lar** least-angle regression **stepwise** forward stepwise regression but see the details section below for other possibilities. Default is `"lasso"`.
`removeColumns`	A logical scalar indicating whether columns with small variance should be removed from consideration as predictors; default 'TRUE'.
`eps`	Numerical tolerance used for assessment of sign, equality, rank determination, column removal, etc. The default is the square root of `.Machine$double.eps`, the relative machine precision.
`blockSize`	If `NULL`, the block size is determined using methods of class `ff`. Otherwise the passed value is used.
`maxStages`	The maximum number of stages allowed in the algorithm. This argument applies only to the `lasso` option. The defaults is `2p` for `lasso`, where `p` if the number of predictors.

Details

An intercept is always included in the regression. This function calls other routines to do the core calculations, one of `biglars.fit.lasso`, `biglars.fit.lar`, or `biglars.fit.stepwise`. These functions are associated with Fraley et~al. (2007) and will not be undergoing further development except for things like bug fixes. For ongoing development of least-angle regression, see the `glars` library.

Value

A list with the following elements:

coef	An array of regression coefficients for each stage.
moves	Any array describing variables added or removed at each stage.
RSS	Residual sum of squares.

References

B. Efron, T. Hastie, I. Johnstone and R. Tibshirani (2004), "Least Angle Regression" (with discussion), *Annals of Statistics* **32**, 407-499.

C. Fraley and T. Hesterberg (2007), " Least-Angle Regression for Large Datasets", Technical Report, Insightful Corporation.

See Also

qrBlockApply

Examples

```
data(diabetes)

larFit <- biglars.fit(diabetes$x, diabetes$y, type = "lar")
larFitBlocked <- biglars.fit(diabetes$x, diabetes$y, type = "lar",
                           blockSize = 50)

lassoFit <- biglars.fit(diabetes$x, diabetes$y, type = "lasso")
lassoFitBlocked <- biglars.fit(diabetes$x, diabetes$y, type = "lasso",
                               blockSize = 34)
```

---

diabetes *Blood and other measurements in diabetics*

Description

The diabetes1 data frame has 442 rows and 1 columns. These are the data used in the Efron et al "Least Angle Regression" paper.

The diabetes data frame has 442 rows and 3 matrices, containing predictors, response, and interactions.

Format

`diabetes1` is a data frame with 442 observations on the following 11 variables.

`age` a numeric vector

`sex` a numeric vector

`bmi` a numeric vector

`map` a numeric vector

`tc` a numeric vector

`ldl` a numeric vector

`hdl` a numeric vector

`tch` a numeric vector

`ltg` a numeric vector

`glu` a numeric vector

`y` a numeric vector

In the `sex` variable, 1 indicates female and 2 male.

`diabetes` is a data frame containing the following objects:

x a matrix with 10 columns–the first 10 columns from `diabetes1`, standardized

y a numeric vector

x2 a matrix with 64 columns–main effects and second-order interactions

Details

The `x` matrix is standardized to have unit L2 norm in each column and zero mean. The matrix `x2` consists of `x` plus second-order powers and interactions, also standardized.

Source

http://www-stat.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.ps

References

B. Efron and T. Hastie (2003), "LARS software for R and Splus", http://www-stat.stanford.edu/~hastie/Papers/LARS

B. Efron, T. Hastie, I. Johnstone and R. Tibshirani (2004), "Least Angle Regression" (with discussion), *Annals of Statistics* **32**, 407-499.

Examples

```
data(diabetes)

stepFit <- biglars.fit(diabetes$x, diabetes$y, type = "stepwise")
stepFitBlocked <- biglars.fit(diabetes$x, diabetes$y, type = "stepwise",
                              blockSize = 50)

lassoFit <- biglars.fit(diabetes$x, diabetes$y)
lassoFitBlocked <- biglars.fit(diabetes$x, diabetes$y, blockSize = 34)
```

qrBlockApply *Blockwise Cholesky Factorization*

Description

Cholesky factorization of a crossproduct matrix via blockwise orthogonal transformation.

Usage

```
qrBlockApply(x, y = NULL, blockSize = NULL)
```

Arguments

x	A numeric matrix or ff. The number of columns should not be larger than the number of rows.
y	An optional numeric vector or 1 column ff of responses.
blockSize	This is used for setting the block size for sequential access. The default is determined by methods from class ff.

Value

A list with the following elements:

R	Upper triangular Cholesky factor of crossprod(x).
Qty	The corresponding transformation of y.

References

C. Fraley and T. Hesterberg (2007), "Least-Angle Regression for Large Datasets", Technical Report, Insightful Corporation.

See Also

biglars.fit

Examples

```
data(diabetes)

qrx <- qr(diabetes$x)

list(R = qr.R(qrx), Qty = crossprod(qr.Q(qrx), diabetes$y))

qrBlockApply( diabetes$x, diabetes$y, blockSize = 34)
```

Index

∗Topic **datasets**
diabetes, 3
∗Topic **regression**
biglars.fit, 2
qrBlockApply, 5

biglars.fit, 2, *5*

diabetes, 3
diabetes1 *(*diabetes*)*, 3

qrBlock *(*qrBlockApply*)*, 5
qrBlockApply, *3*, 5